

Advant[®] OCS
with Master software

Advant Object Library for Advant Controllers

Version 1.3

Reference Manual

3BSE 013 934R301



Use of **DANGER**, **WARNING**, **CAUTION**, and **NOTE**

This publication includes, **DANGER**, **WARNING**, **CAUTION**, and **NOTE** information where appropriate to point out safety related or other important information.

DANGER	Hazards which could result in severe personal injury or death
WARNING	Hazards which could result in personal injury
CAUTION	Hazards which could result in equipment or property damage
NOTE	Alerts user to pertinent facts and conditions

Although **DANGER** and **WARNING** hazards are related to personal injury, and **CAUTION** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **DANGER**, **WARNING**, and **CAUTION** notices.

TRADEMARKS

Advant, AdvaCommand, AdvaInform, and AdvaBuild are registered trademarks of ABB Asea Brown Boveri Ltd., Switzerland.

NOTICE

The information in this document is subject to change without notice and should not be construed as a commitment by ABB Automation Products GmbH. ABB Automation Products GmbH assumes no responsibility for any errors that may appear in this document.

In no event shall ABB Automation Products GmbH be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB Automation Products GmbH be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without ABB Automation Products GmbH's written permission, and the contents thereof must not be imparted to a third party nor be used for any unauthorized purpose.

The software described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license.

Copyright © ABB Automation Products GmbH 1999

TABLE OF CONTENTS

Chapter 1 - Introduction

1.1	Before You Begin.....	1-1
1.2	Equipment Requirements.....	1-1
1.3	How to Use This Book.....	1-1
1.4	Conventions	1-2
1.5	Related Documentation.....	1-3
1.6	Release History	1-4

Chapter 2 - Installation

Chapter 3 - Content of the Advant Object Library for Advant Controllers

3.1	Introduction.....	3-1
3.2	Advant Objects Representing Functions.....	3-1
3.2.1	Content of the Function Structure.....	3-1
3.2.1.1	Advant Objects Assigned to Control Loops.....	3-2
3.2.1.2	Advant Objects Assigned to Drive Systems.....	3-3
3.2.1.3	Advant Objects Assigned to Motorfunction	3-10
3.2.1.4	Advant Objects Assigned to Valves	3-12
3.2.1.5	Advant Objects Representing Signals.....	3-12
3.3	Advant Objects Classified as Control System Items	3-19
3.3.1	Content of the OCS-Structure.....	3-19
3.3.1.1	Advant Objects Assigned to Controllers.....	3-20
3.3.1.2	Structure of Boards and Signals.....	3-21
3.3.1.3	Board Types in the Advant Object Library	3-21
3.3.1.4	Advant Objects Representing Signals.....	3-29
3.4	Advant Objects Classified as Mounting Locations.....	3-29
3.4.1	Content of the Mounting Location Structure.....	3-30

Chapter 4 - Hints and Recommendations

4.1	Version Settings for Controller and Tools in the Object Library	4-1
4.1.1	Presettings for DB Elements for Boards and Signals	4-1
4.1.2	Changing the Version Settings.....	4-1
4.1.2.1	Changing the Controller Version.....	4-1
4.1.2.2	Changing the FCB Version	4-2
4.2	Recommendations.....	4-3
4.2.1	Configuring an Object with Circuit Aspect Containing Type Circuit	4-3
4.2.2	Configuring an Object to Create DB Elements by Drag and Drop.....	4-3

CONTENTS (continued)

4.2.3	Configuring an Object to Create DB Elements by Generate DB.....	4-4
4.2.4	Setting Options for Bulk Copy From Library	4-4
4.2.5	DBElement Aspect Copied from Library	4-4
4.2.6	Prepare Structure Aspects	4-5
4.2.7	Environment Variables TC\$LEV1 and TC\$LEV2.....	4-5

ILLUSTRATIONS

Figure 3-1.	Basic Function Oriented Structure of the Advant Object Library	3-1
Figure 3-2.	Advant Object Types Representing Control Loops	3-2
Figure 3-3.	Advant Objects Representing Drive Systems	3-3
Figure 3-4.	Aspects of Drive System DS.C1	3-4
Figure 3-5.	Aspects of Drive System DS.P1	3-5
Figure 3-6.	Form View of Electrical Structure Aspect Data	3-6
Figure 3-7.	Datasheet View of Electrical Device Aspects.....	3-6
Figure 3-8.	Datasheet View of Electrical Function Aspect	3-7
Figure 3-9.	Typical Circuit Diagrams in the Binder Objects of DS.P1	3-8
Figure 3-10.	DB-Section Window of Function Chart Builder.....	3-9
Figure 3-11.	Signal Parameters	3-9
Figure 3-12.	Advant Objects Representing Motor Functions.....	3-10
Figure 3-13.	Example of a Circuit Assigned to the Advant Object M_DR.....	3-11
Figure 3-14.	Example of a Parameter Aspect Type	3-12
Figure 3-15.	Advant Objects Representing Control Valve Functions	3-12
Figure 3-16.	Signals for S100 I/O Modules	3-13
Figure 3-17.	Function Chart View on the DB Element Aspect of an AIS Signal	3-14
Figure 3-18.	Source Code View on the DB Element Aspect of an AIS Signal.....	3-15
Figure 3-19.	Attribute View on a Signal Parameter Aspect	3-16
Figure 3-20.	Attribute View on the El. Function Aspect for Signals.....	3-16
Figure 3-21.	Signals for S400 I/O Modules	3-17
Figure 3-22.	Signals for S600 I/O Modules	3-17
Figure 3-23.	Signals for S800 I/O Modules	3-18
Figure 3-24.	Basic Control System Oriented Structure of the Advant Object Library ...	3-19
Figure 3-25.	Advant Object Types Representing Controllers, Bus Systems and Stations	3-20
Figure 3-26.	Structure of Boards and Signals.....	3-21
Figure 3-27.	S100 I/O Boards.....	3-22
Figure 3-28.	Function Chart View on a Board DB Element.....	3-23
Figure 3-29.	Board Parameters.....	3-23

ILLUSTRATIONS (continued)

Figure 3-30.	Source Code of a Board DB Element.....	3-24
Figure 3-31.	Attribute View on an El. Device Aspect of a Board.....	3-24
Figure 3-32.	Diagram View on the El. Circuit Diagram Aspect of Board DSDI110A...	3-25
Figure 3-33.	Presentation of Board Channels	3-26
Figure 3-34.	Attribute View on an El. Function Aspect of a Board.....	3-27
Figure 3-35.	S400 I/O Boards	3-27
Figure 3-36.	S600 I/O Boards	3-28
Figure 3-37.	S800 I/O Boards	3-29
Figure 3-38.	Basic Mounting Location Oriented Structure of the Object Library.....	3-30
Figure 3-39.	OCS-Cabinet Assemblies	3-30
Figure 3-40.	Racks and Board Locations in an OCS-Cabinet.....	3-31
Figure 3-41.	Example for a S800 Mounting Unit	3-32
Figure 3-42.	Example for Motor Controller Cabinet	3-32
Figure 4-1.	Modify Type Circuit Window.....	4-2
Figure 4-2.	Select Tool Versions Box.....	4-2

TABLES

Table 1-1.	Related Documentation	1-3
Table 1-2.	Versions of the Object Library for Advant Controllers	1-4
Table 1-3.	Versions of this Reference Manual.....	1-4

Chapter 1 Introduction

1.1 Before You Begin

This Reference Manual contains information about the Advant Object Library for Advant Controllers. The Object Library shows predefined Advant object types that can be used as examples by copy or by import procedures of the AdvanBuild Engineering Software. It also shows as an example how to structure a library of Advant objects for project specific applications.

The Object library contains DB Element aspects, Type Circuit aspects and Circuit aspects. The content of separate libraries is used to create these aspects, that are the DB Element and PC Element libraries. These libraries are accessed by the Function Chart builder for the creation and application of the Object library.

An Example Project that is delivered in conjunction with the Object library gives a short overview about the structure, the content and the principles of a project model that is created by applying the Advant object concept.

1.2 Equipment Requirements

For a description of the system requirements consult the release Notes of AdvanBuild Engineering Workplace and the Release Notes of the individual Builders.

1.3 How to Use This Book

This book is organized into the following chapters:

Chapter 1 - Introduction

Chapter 2 - Installation

Chapter 3 - Content of the Object Library

Chapter 4 - Hints and Recommendations.

1.4 Conventions

Character Style

- Use of *italic* characters
 - to emphasize a term or phrase
 - to indicate variables
Example: the files are in the **data\project\dob\work** directory
 - to reference titles of manuals
Example: See the *Structure Builder User's Guide* for information about.....
- Use of **bold** characters
 - for menu items
Example: Choose **Save** from the **File** menu (alternative: Choose **File** | **Save**)
 - for names of windows and boxes
Example: Press **Add...** in the **Control_Valve_1** window or Press **Add...** in the **Object** window
 - for names of files, directories and paths
Example: the files are in the **data\exlib\dob\work** directory.

1.5 Related Documentation

Table 1-1 lists the documentation related to the content of this reference manual:

Table 1-1. Related Documentation

Title	Description
Structure Builder User's Guide	This book describes the creation of projects, engineering objects, structures and aspects and how to navigate in projects
AMPL Control Configuration Commands for Structure Builder User's Guide	This book describes the cooperation of the Application Builder, Function Chart Builder, On-line Builder and Structure Builder
Application Builder User's Guide	This book describes the installation of functions and components for control configuration and contains the operating instructions of Application Builder, Bus Configuration Builder and support functions
Function Chart Builder User's Guide	This book contains a description of AMPL application programs configuration
Parameter Builder User's Guide	This book describes how to handle parameters for process objects and for OCS objects and how to import engineering data
Document Builder User's Guide	This book describes the support of handling various types of documents, including the handling of templates
Electrical Diagram Builder User's Guide	This book describes the electrical plant engineering and the handling of circuit diagrams
Wiring Builder User's Guide	Describes how to create and edit electrical wiring data of wires and cables assigned to physical connections between devices.
Engineering Methods Users Guide	This book describes the principles of engineering based on the Advant object concept

1.6 Release History

This is the third release of the Advant Object Library for Advant Controllers.

Table 1-2. Versions of the Object Library for Advant Controllers

Version	Release Date	Remarks
Advant Object Library for Advant Controllers 1.0/0	12-Dec-1997	First version
Advant Object Library for Advant Controllers 1.1/0	31-July-1998	Second version
Advant Object Library for Advant Controllers 1.2/0	26-Feb-1999	Third version
Advant Object Library for Advant Controllers 1.3/0	July 1999	Fourth version

Table 1-3. Versions of this Reference Manual

Version	Description
3BSE013934R0001	Describes the content of the Advant Object Library for Advant Controllers 1.0/0
3BSE013934R0101	Describes the content of the Advant Object Library for Advant Controllers 1.1/0
3BSE013934R201	Describes the content of the Advant Object Library for Advant Controllers 1.2/0
3BSE013934R301	Describes the content of the Advant Object Library for Advant Controllers 1.3/0

Chapter 2 Installation

See Advant Object Library for Advant Controllers Release Notes.

Chapter 3 Content of the Advant Object Library for Advant Controllers

3.1 Introduction

The Advant Object Library contains predefined Advant object types which can be used as solution examples for demonstrations. It also shows how to structure an Advant object library. You can use the Advant object examples representing OCS Boards and OCS Signals for application examples when you parametrize it. Library contents are used for an application by copying it directly or via an import procedure.

The degree of specification of the Advant objects in the library is different. Some objects are specified for their use as complete examples, others are only used for structuring and grouping purpose and carry only a basic property information and structure information.

3.2 Advant Objects Representing Functions

Advant objects representing functions or viewed as functions are presented in a function oriented structure built up by the Structure Builder. The Advant objects are assigned to a basic structure that groups the Advant objects according to the type of function they represent. The presented basic structure has to be regarded as an example that can be modified and expanded by the user.

3.2.1 Content of the Function Structure

Figure 3-1 shows the basic structure to which Advant objects that mainly represent functions will be assigned:

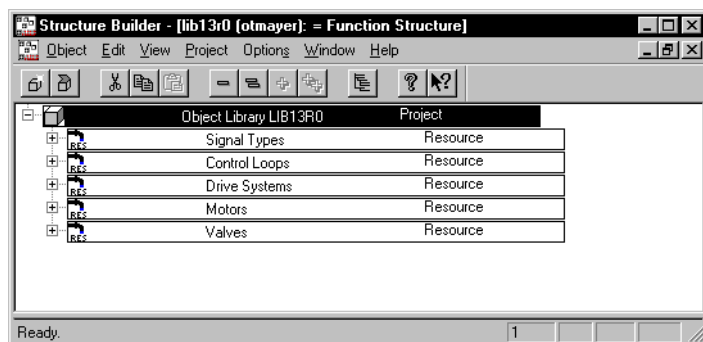


Figure 3-1. Basic Function Oriented Structure of the Advant Object Library

3.2.1.1 Advant Objects Assigned to Control Loops

Figure 3-2 shows the Advant object examples that are assigned to control loops of the basic structure.

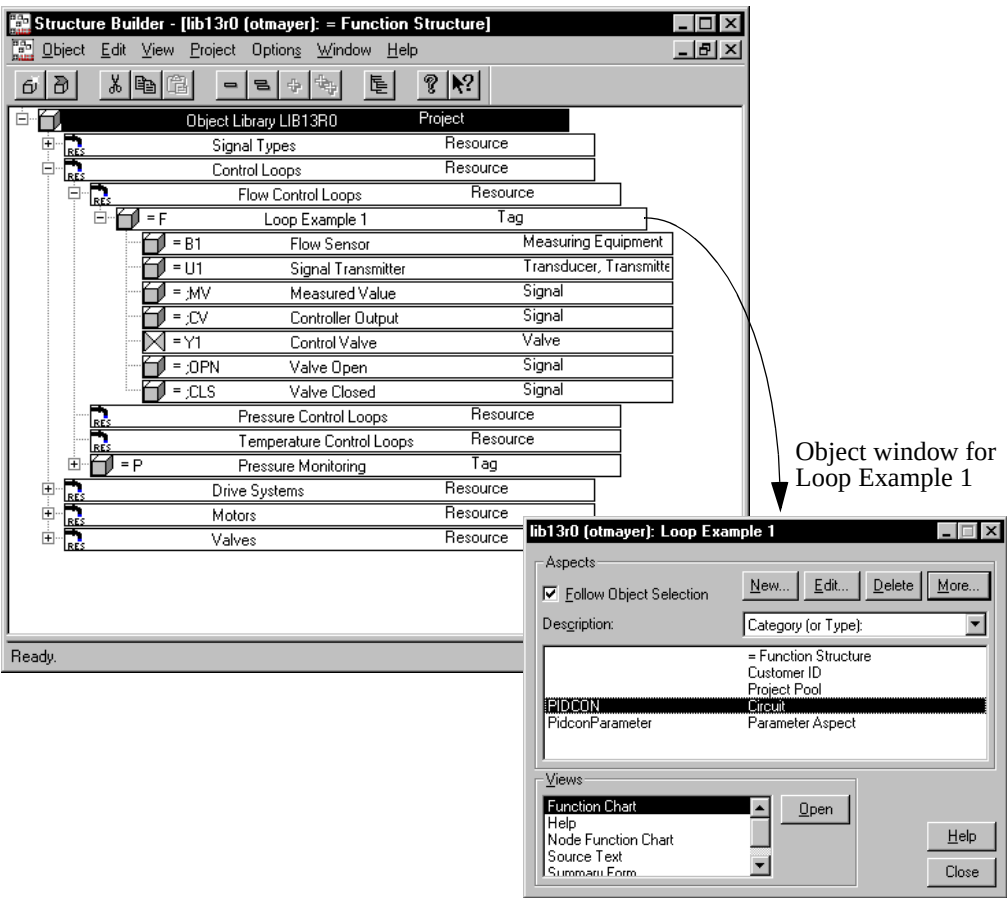


Figure 3-2. Advant Object Types Representing Control Loops

The tag object represents a complete control loop. When such a control loop is used in a project additional Document aspects as for example a Tag Specification Sheet and an Instrumentation List are usually assigned to the tag object.

3.2.1.2 Advant Objects Assigned to Drive Systems

Drive systems include objects as motor, motorstarter, input/output signals and so on with all the information needed for a complete drive function. Such groups of objects representing drive systems are assigned to the corresponding structure object named Drive Systems.

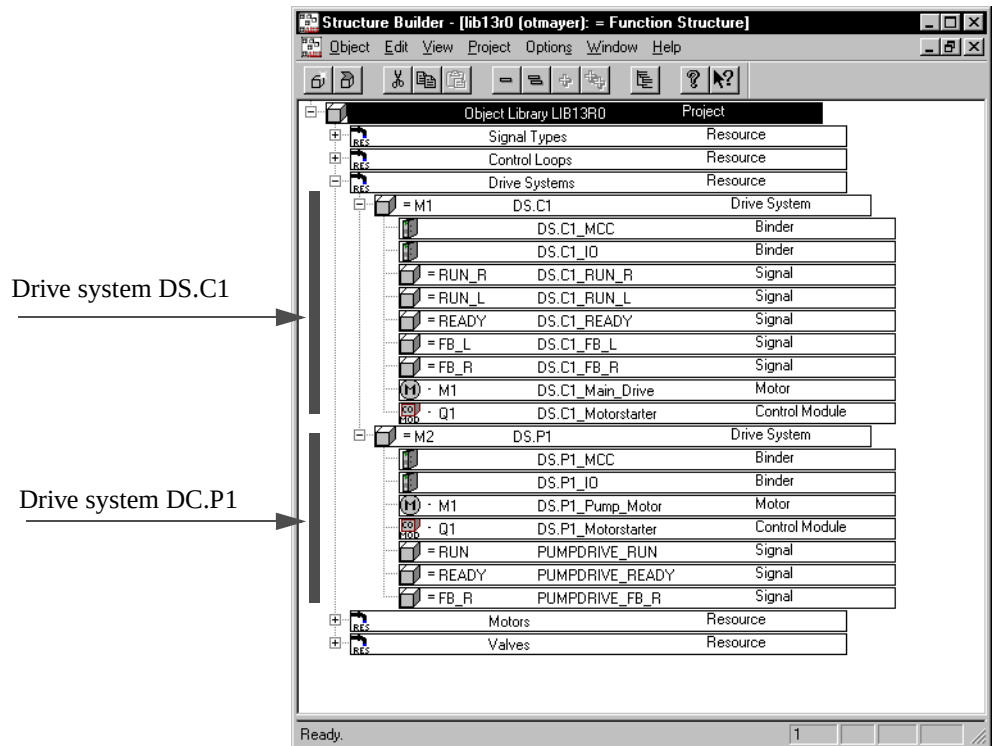


Figure 3-3. Advant Objects Representing Drive Systems

Two drive systems are available as examples in the Object Library. One drive system named DS.C1 for a two direction drive with AC motor and one drive system named DS.P1 for a pump drive with AC motor. Figure 3-4 and Figure 3-5 show the aspects assigned to these drive systems in the aspect area of the Structure Builder window. (Basic property aspects and structure aspects of the Structure Builder are filtered out and not shown.)

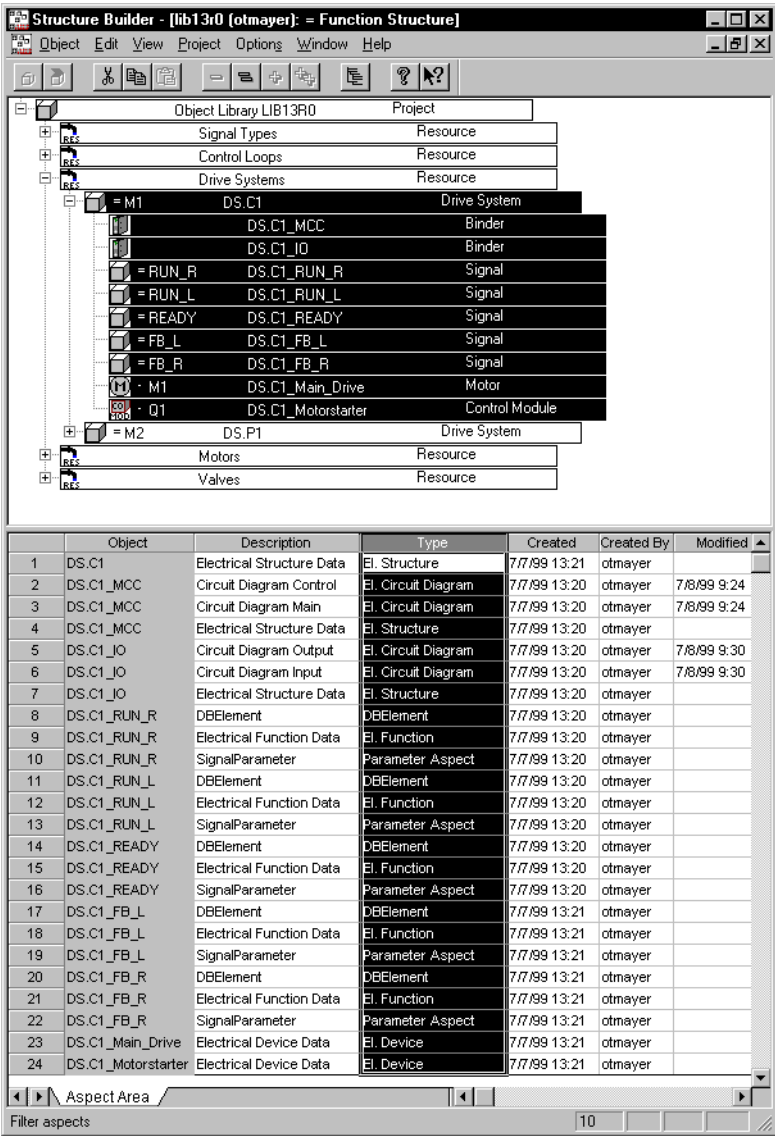


Figure 3-4. Aspects of Drive System DS.C1

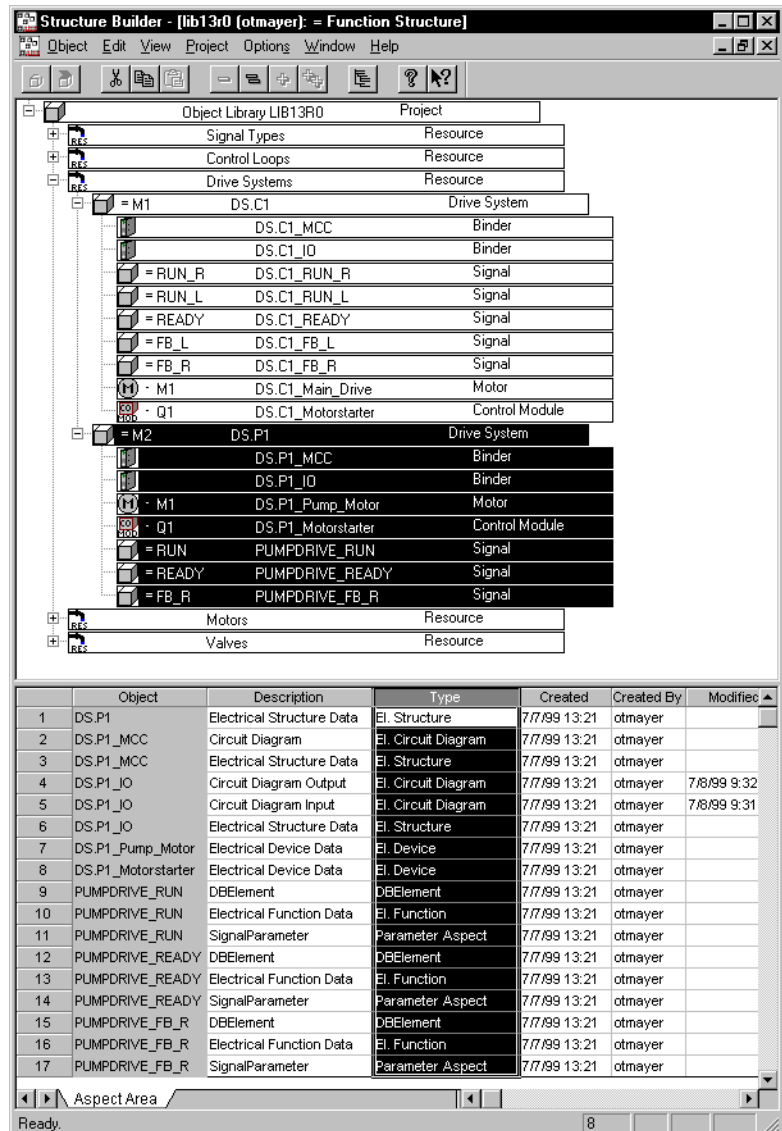


Figure 3-5. Aspects of Drive System DS.P1

These drive systems contain Advant object aspects handled by the Electrical Diagram Builder. These are aspects, which can be connected and linked to symbols in diagrams and therefore exchange data between database and symbols in diagrams. The two drive systems contain the following aspects belonging to the group *electrical plant object aspects*:

Electrical Structure Aspect

This aspect type is set in Advant objects of the drive systems where no other electrical plant object aspect is inserted. Figure 3-6 shows the Form view of the Electrical Structure aspect of object DS.P1.

Figure 3-6. Form View of Electrical Structure Aspect Data

Electrical Device Aspect

This aspect type is inserted in Advant objects that represent an electrical device. The Advant objects representing motors and the motorstarters carry this aspect in our example. Figure 3-7 shows the datasheet view of all electrical device aspects in the two drive system examples.

Prod.Ref.Des:	Loc.Ref.Des:	Func.Ref.Des.:	Category:	OCS Item Des.:	Free Des.:	Des.Ru
=M1-M1		=M1	MOTOR			ID1
=M1-Q1		=M1	MOTOR_STARTEF			ID1
=M2-M1		=M2	MOTOR			ID1
=M2-Q1		=M2	MOTOR_STARTEF			ID1

Figure 3-7. Datasheet View of Electrical Device Aspects

Electrical Function Aspect

This aspect type is inserted in Advant objects that represent signals and channels. Figure 3-8 shows the datasheet view of all electrical function aspects in the two drive systems of our example.

Prod.Ref.Des:	Loc.Ref.Des:	Func.Ref.Des:	Des.Rule:	Denom.1:	Category:	AspectType:	De
		=M1_FB_L	ID1		SIGNAL	EL_FUNC	
		=M1_FB_R	ID1		SIGNAL	EL_FUNC	
		=M1_READY	ID1		SIGNAL	EL_FUNC	
		=M1_RUN_L	ID1		SIGNAL	EL_FUNC	
		=M1_RUN_R	ID1		SIGNAL	EL_FUNC	
		=M2_FB_R	ID1	Feedback Running	SIGNAL	EL_FUNC	
		=M2_READY	ID1	Ready for Operation	SIGNAL	EL_FUNC	
		=M2_RUN	ID1	Run Pump	SIGNAL	EL_FUNC	

Record: 1 of 8 (Filtered)

Figure 3-8. Datasheet View of Electrical Function Aspect

Electrical Circuit Diagram Aspect

Electrical Circuit Diagram aspects are assigned to Advant objects of type Binder. Electrical Circuit Diagrams in the library contain typicals that are completed and interconnected in the application in which they will be used. In the standard pump drive system of our example we have several typical circuit diagrams, one for the main motor circuit and the motorstarter logic, one for the OCS-input signals circuit diagram and one for the OCS-output signals circuit diagram. Figure 3-9 shows the typical for the main motor circuit and a typical for the input signals circuit and output signal circuit of the drive system example DS.P1.

Drive system DS.C1 contains in its binder objects also typical circuit diagrams of the motor feed, of the motorstarter control logic and of the input/output signals.

Example of a typical circuit diagram for motor feed and motorstarter
 (in object DS.P1_MCC)

Example of a typical circuit diagrams for OCS inputs and outputs
 (in object DS.P1_IO)

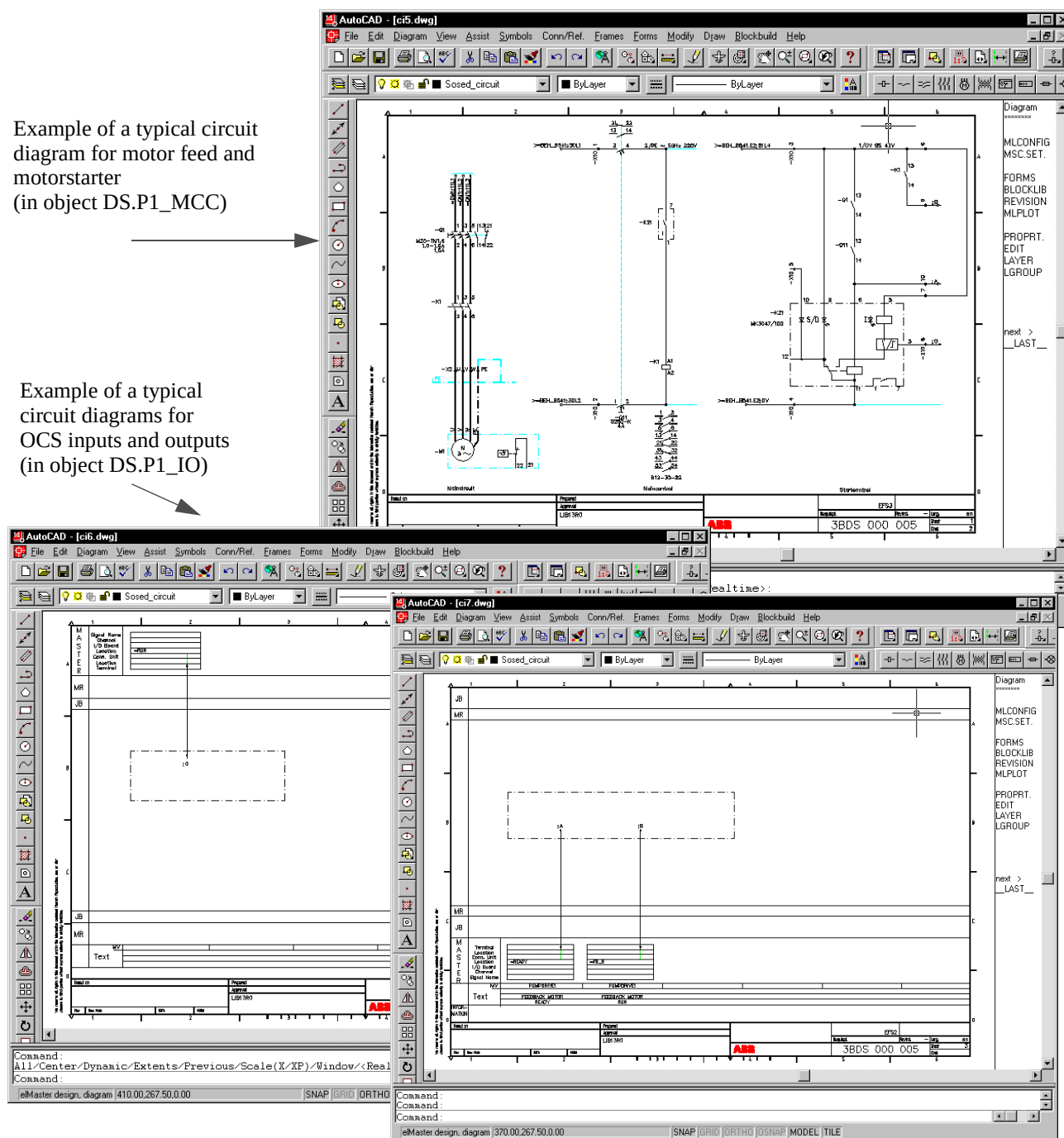


Figure 3-9. Typical Circuit Diagrams in the Binder Objects of DS.P1

The drive system examples contain besides the electrical plant object aspects the following aspects

- Database Element Aspects

DBElement aspects are assigned to the I/O signals of the drive systems and reference to a type circuit for the corresponding signal DB Element. Figure 3-10 shows an example in the DB section window of the Function Chart Builder.

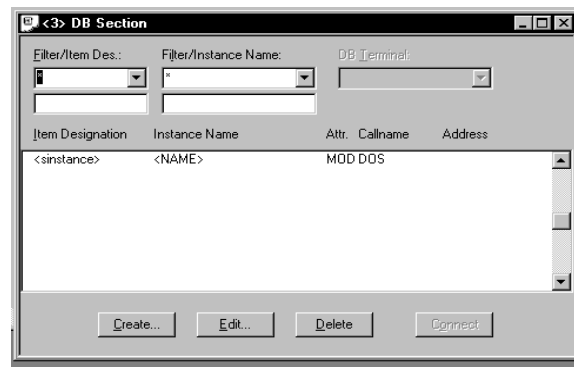


Figure 3-10. DB-Section Window of Function Chart Builder

- Parameter Aspects

Figure 3-11 shows a predefined parameter aspect used for the signal objects in the drive systems.

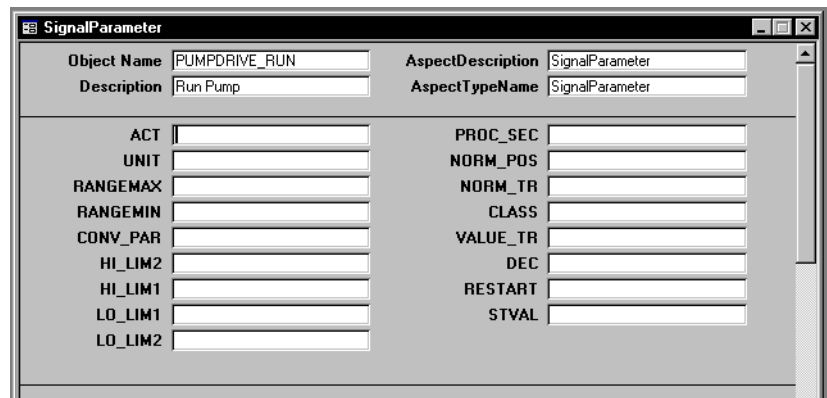


Figure 3-11. Signal Parameters

- Circuit Aspect

The top Advant object of the subtree representing the standard drive system would carry in complete standards a circuit aspect with the control logic for the defined standard. This circuit aspect is not inserted in the presented examples to keep the focus on the electrical engineering.

3.2.1.3 Advant Objects Assigned to Motorfunction

Objects assigned to the group of motors are drive systems that are only handled with respect to its OCS Functions and which are represented by one object only for this purpose. No other equipment as for example in a drive system is assigned. The Advant object M_DR, M_KL and MOT1L contain each a circuit aspect for the control logic and can be used for demonstration purpose as for example for the import function.

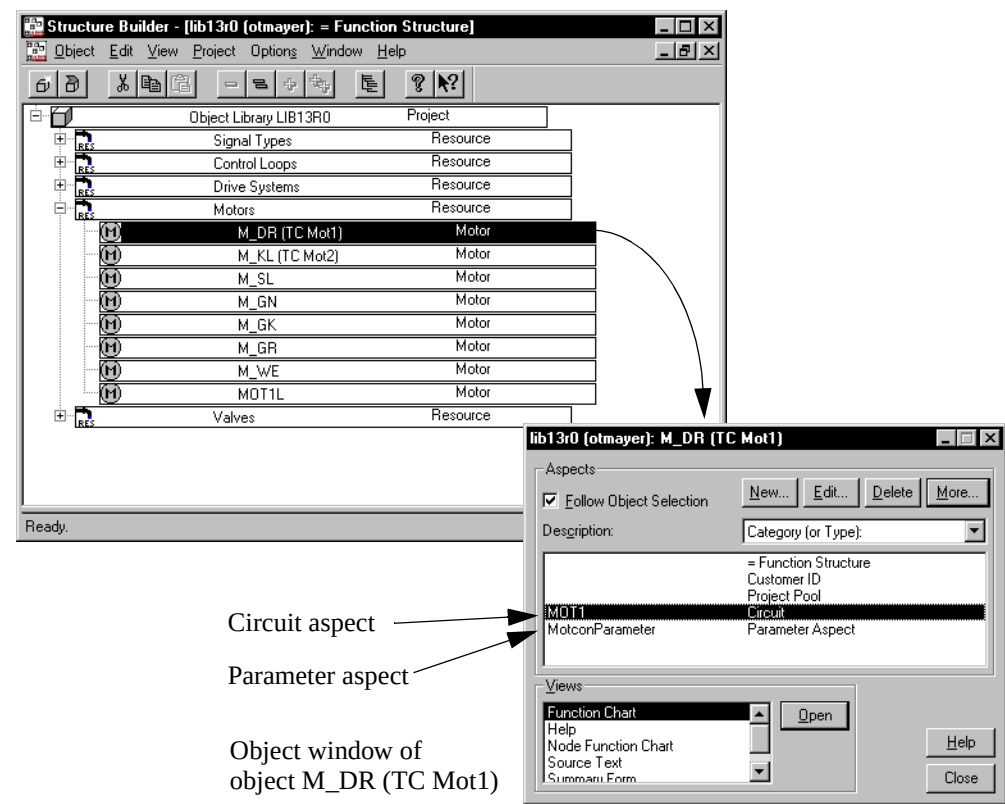


Figure 3-12. Advant Objects Representing Motor Functions

Figure 3-13 shows the Function Chart view of the Circuit aspect in object M_DR with the DB Section, PC Section and Symbol Section window.

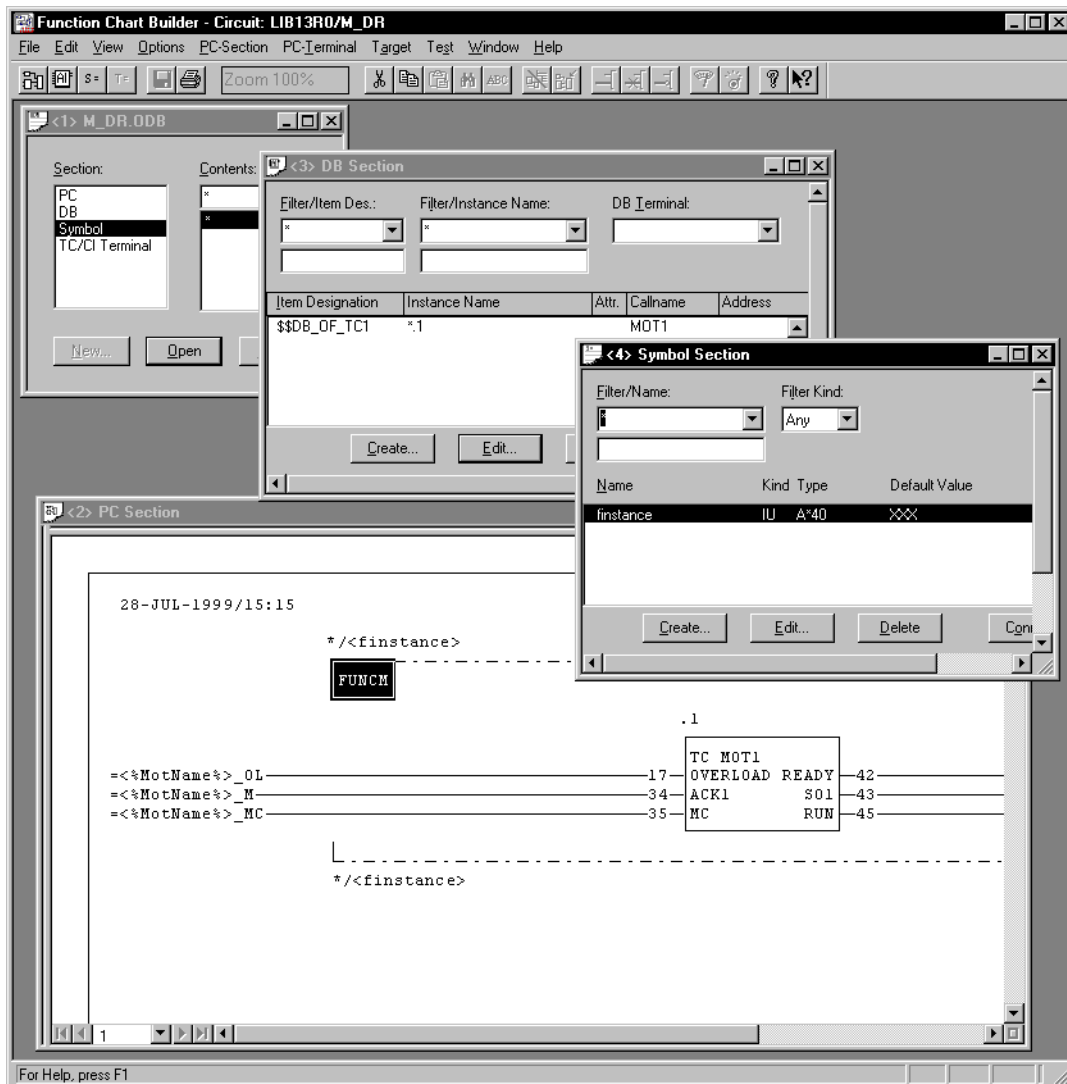


Figure 3-13. Example of a Circuit Assigned to the Advant Object M_DR

Figure 3-14 shows the parameters defined in a Parameter aspect named MotconParameter.

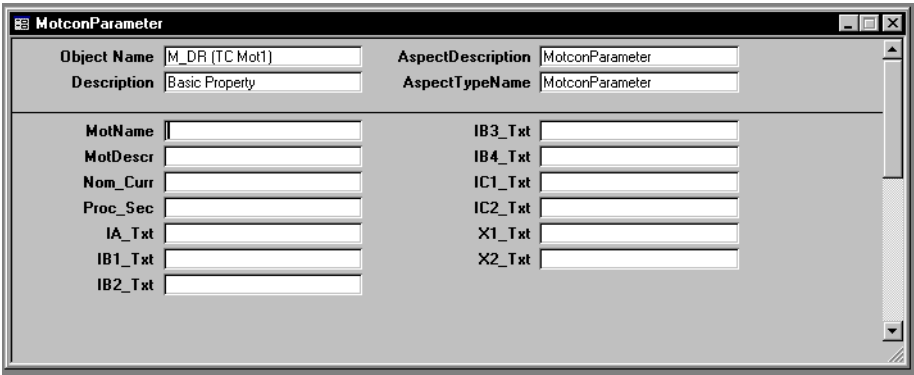


Figure 3-14. Example of a Parameter Aspect Type

The parameter values in parameter aspects of the Advant Object Library are not filled in. The values are input when the objects are used for applications. Parameter values are used by Type Circuits within Circuits. Default values for parameters are defined in the Type Circuit specifications.

3.2.1.4 Advant Objects Assigned to Valves

Figure 3-15 is an example for the grouping of control valve functions in the library. They are put into the function structure if the OCS part is the main focus of this objects.

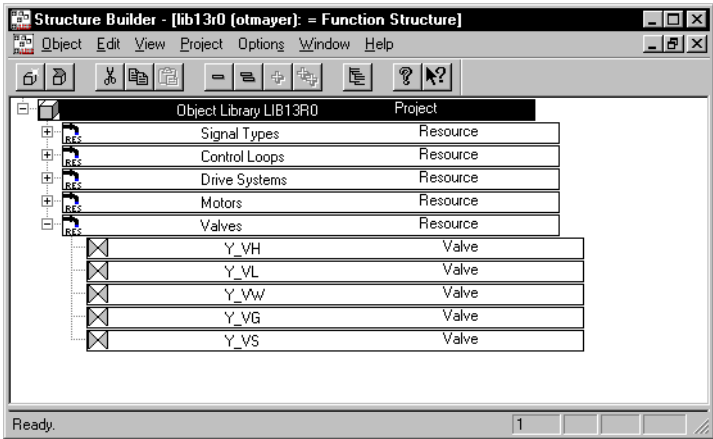


Figure 3-15. Advant Objects Representing Control Valve Functions

3.2.1.5 Advant Objects Representing Signals

Signals are classified in the Advant Object Library according to their belonging to the I/O modules
S100, S400, S600, S800.

DB Element aspects and Signal Parameter aspects are assigned to all signal objects in the Advant Object Library.

Signal objects are specified with all aspects necessary for their use as examples in projects, for example by the import function.

Signal objects are placed in two library structures, the function oriented and the control system oriented structure. Doing this, signals get the needed structure aspects when used by the import function for a target project.

Signals for S100 I/O modules

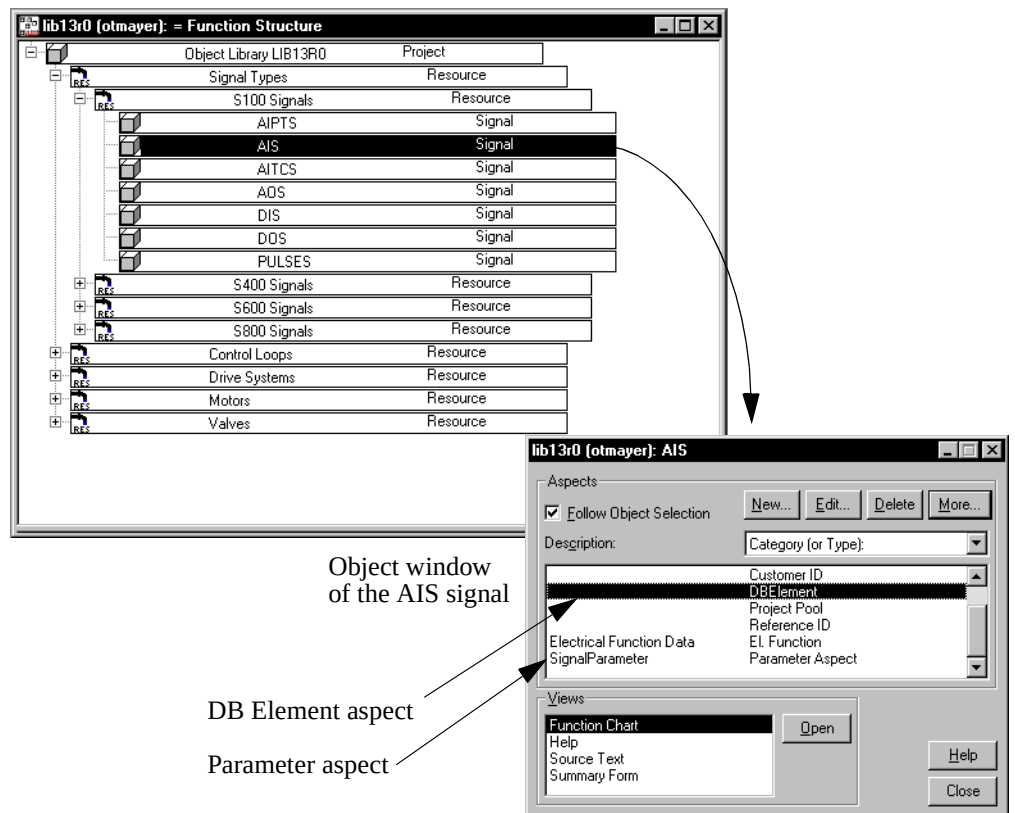


Figure 3-16. Signals for S100 I/O Modules

Figure 3-17 shows the Function Chart view on the DB aspect of an Advant object representing a signal. The figure shows the DB Section and the Symbol Section window.

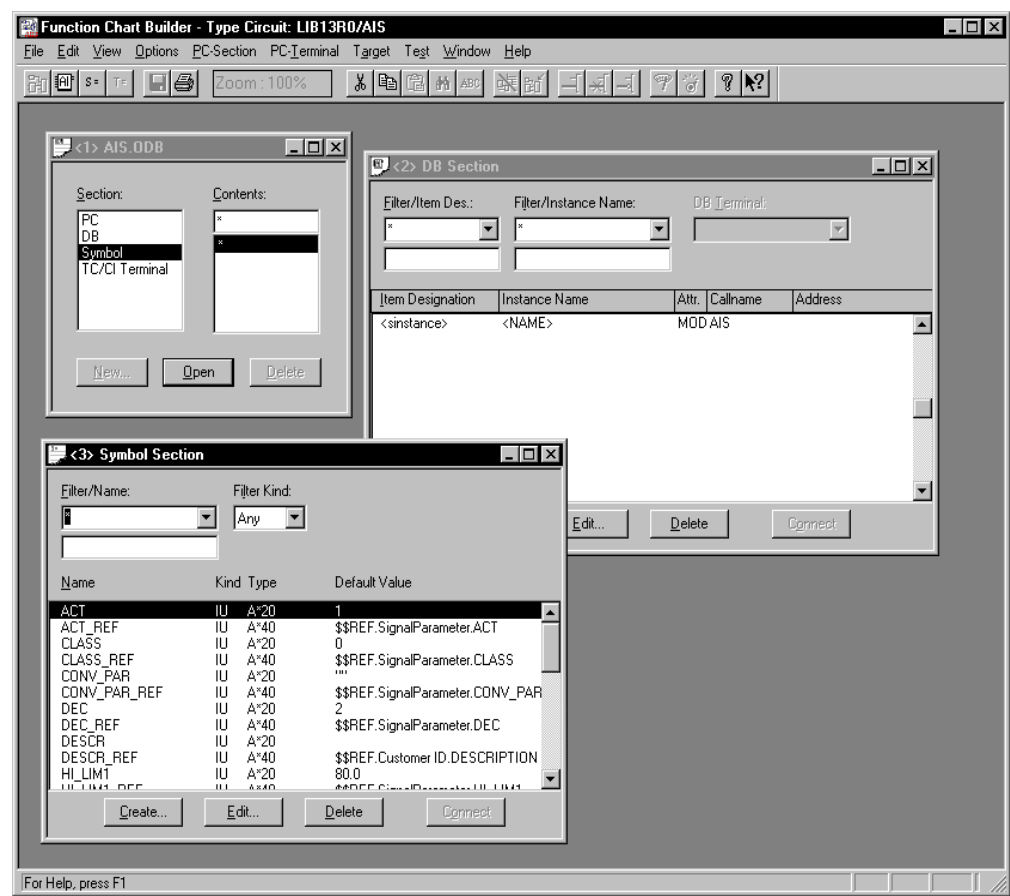
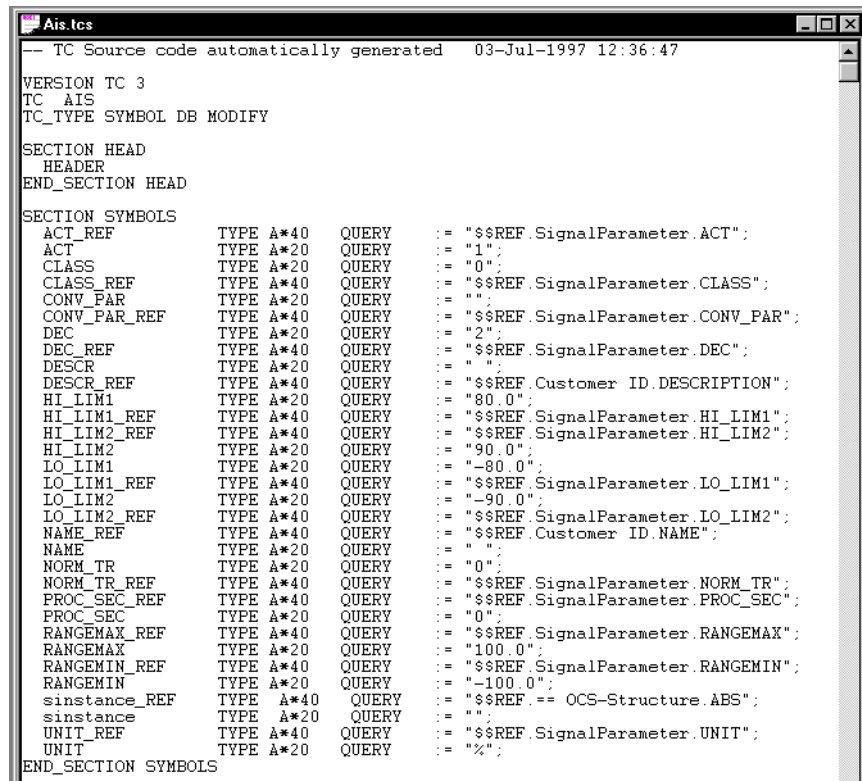


Figure 3-17. Function Chart View on the DB Element Aspect of an AIS Signal

Figure 3-18 shows the Source code of the DB Element aspect of a Signal:



```

Ais.tcs
-- TC Source code automatically generated 03-Jul-1997 12:36:47

VERSION TC 3
TC AIS
TC_TYPE SYMBOL DB MODIFY

SECTION HEAD
HEADER
END_SECTION HEAD

SECTION SYMBOLS
ACT_REF          TYPE A*40  QUERY := "$$REF.SignalParameter.ACT";
ACT              TYPE A*20  QUERY := "1";
CLASS            TYPE A*20  QUERY := "0";
CLASS_REF        TYPE A*40  QUERY := "$$REF.SignalParameter.CLASS";
CONV_PAR         TYPE A*20  QUERY := "";
CONV_PAR_REF     TYPE A*40  QUERY := "$$REF.SignalParameter.CONV_PAR";
DEC              TYPE A*20  QUERY := "2";
DEC_REF          TYPE A*40  QUERY := "$$REF.SignalParameter.DEC";
DESCR            TYPE A*20  QUERY := "";
DESCR_REF        TYPE A*40  QUERY := "$$REF.Customer ID.DESCRPTION";
HI_LIM1          TYPE A*20  QUERY := "80.0";
HI_LIM1_REF      TYPE A*40  QUERY := "$$REF.SignalParameter.HI_LIM1";
HI_LIM2_REF      TYPE A*40  QUERY := "$$REF.SignalParameter.HI_LIM2";
HI_LIM2          TYPE A*20  QUERY := "90.0";
LO_LIM1          TYPE A*20  QUERY := "-80.0";
LO_LIM1_REF      TYPE A*40  QUERY := "$$REF.SignalParameter.LO_LIM1";
LO_LIM2          TYPE A*20  QUERY := "-90.0";
LO_LIM2_REF      TYPE A*40  QUERY := "$$REF.SignalParameter.LO_LIM2";
NAME_REF         TYPE A*40  QUERY := "$$REF.Customer ID.NAME";
NAME             TYPE A*20  QUERY := "";
NORM_TR          TYPE A*20  QUERY := "0";
NORM_TR_REF      TYPE A*40  QUERY := "$$REF.SignalParameter.NORM_TR";
PROC_SEC_REF     TYPE A*40  QUERY := "$$REF.SignalParameter.PROC_SEC";
PROC_SEC         TYPE A*20  QUERY := "0";
RANGEMAX_REF     TYPE A*40  QUERY := "$$REF.SignalParameter.RANGEMAX";
RANGEMAX         TYPE A*20  QUERY := "100.0";
RANGEMIN_REF     TYPE A*40  QUERY := "$$REF.SignalParameter.RANGEMIN";
RANGEMIN         TYPE A*20  QUERY := "-100.0";
sinstance_REF    TYPE A*40  QUERY := "$$REF.== OCS-Structure.ABS";
sinstance        TYPE A*20  QUERY := "";
UNIT_REF         TYPE A*40  QUERY := "$$REF.SignalParameter.UNIT";
UNIT             TYPE A*20  QUERY := "%";
END_SECTION SYMBOLS

```

Figure 3-18. Source Code View on the DB Element Aspect of an AIS Signal

Figure 3-19 shows the Attribute view of the Signal Parameter aspect. This aspect type is assigned to all signal objects in the Advant Object Library. The parameters are inserted when the library objects are used for a project for example by an import procedure.

The screenshot shows a window titled "SignalParameter". It contains two columns of input fields. The left column has fields for ACT, UNIT, RANGEMAX, RANGEMIN, CONV_PAR, HI_LIM2, HI_LIM1, LO_LIM1, and LO_LIM2. The right column has fields for PROC_SEC, NORM_POS, NORM_TR, CLASS, VALUE_TR, DEC, RESTART, and STVAL. At the top, there are fields for Object Name (AIS), Description, AspectDescription (SignalParameter), and AspectTypeName (SignalParameter).

Figure 3-19. Attribute View on a Signal Parameter Aspect

The parameter values in Signal Parameter aspects of the Advant Object Library are not filled in. The values are input when the objects are used for applications. Parameter values are used by the Type Circuits representing DB Elements. The default values for signal parameters are defined in the DB Element specifications.

Figure 3-20 shows the Attribute view of the El. Function aspect. This aspect type is assigned to all signal objects in the Advant Object Library and it is set to category SIGNAL.

The category of the the El. Function aspect for signals must be set to SIGNAL

The screenshot shows a window titled "El.Function: AIS (EL_FUNC : Form)". It has two tabs: "Descriptive Data" and "Free Data". Under "Descriptive Data", there are fields for Func.Ref.Des. (=\$7726), Loc.Ref.Des., Prod.Ref.Des., OCS Item Des., and Free Des. There is also a "Des.Rule:" dropdown set to "ID1". Under "Free Data", there is a "Category:" dropdown set to "SIGNAL", a "Func. Codes:" dropdown, and fields for Denom.1, Denom.2, Denom.3, Remark1, Remark2, and Descr.1. At the bottom, there are buttons for Exit, Save, Delete, and Help, and a status bar showing "Record: 1 of 1 (Filtered)".

Figure 3-20. Attribute View on the El. Function Aspect for Signals

Signals for S400 I/O modules

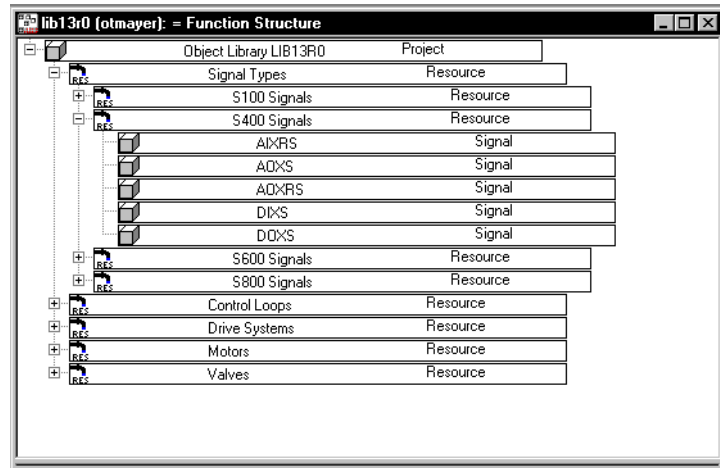


Figure 3-21. Signals for S400 I/O Modules

Signals for S600 I/O modules

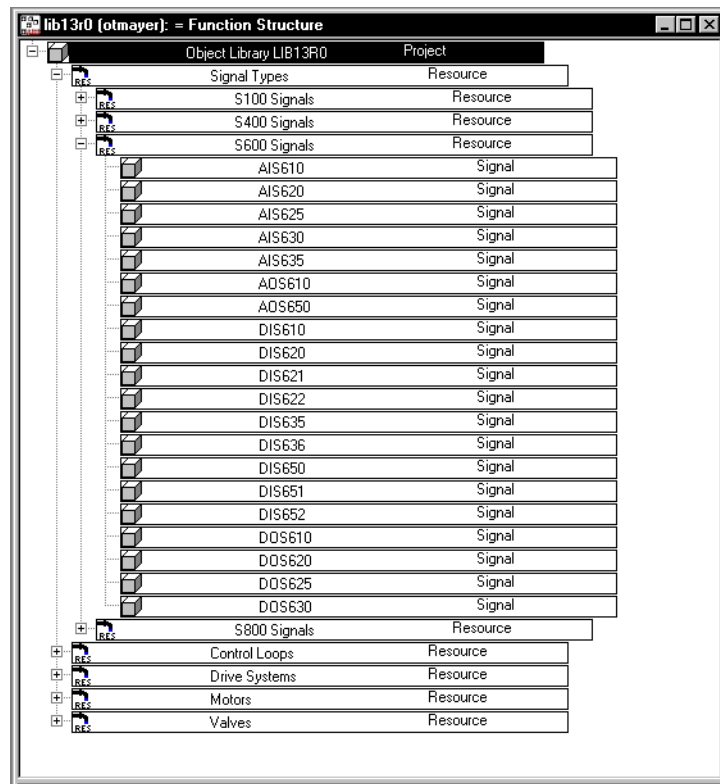


Figure 3-22. Signals for S600 I/O Modules

Signals for S800 I/O modules

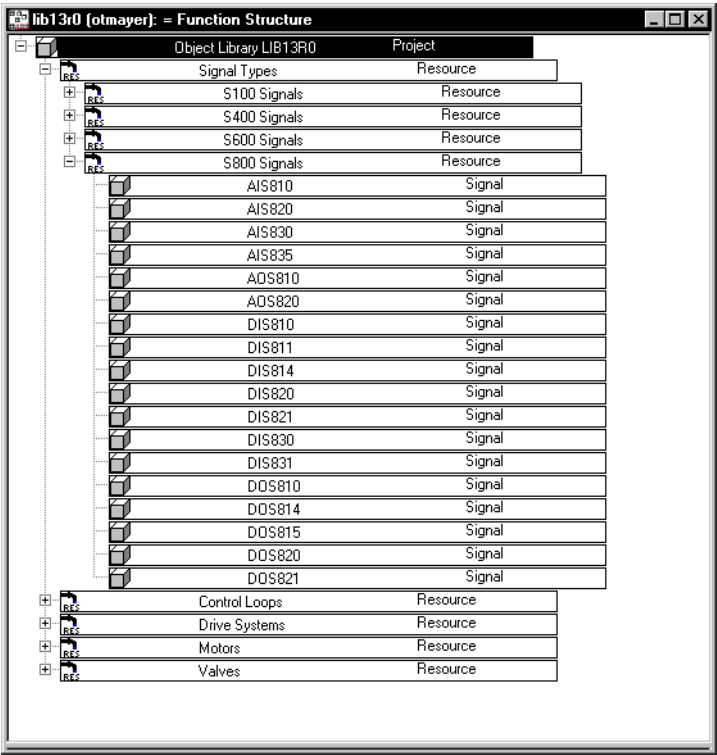


Figure 3-23. Signals for S800 I/O Modules

3.3 Advant Objects Classified as Control System Items

Advant objects representing items of a control system are presented in a control system oriented structure built up by the Structure Builder. The Advant objects are assigned to a basic structure that groups the OCS- objects into several object classes. The presented basic structure has to be regarded as an example that can be modified and expanded by the user.

3.3.1 Content of the OCS-Structure

Figure 3-24 shows the basic structure to which Advant objects that represent control system items are assigned:

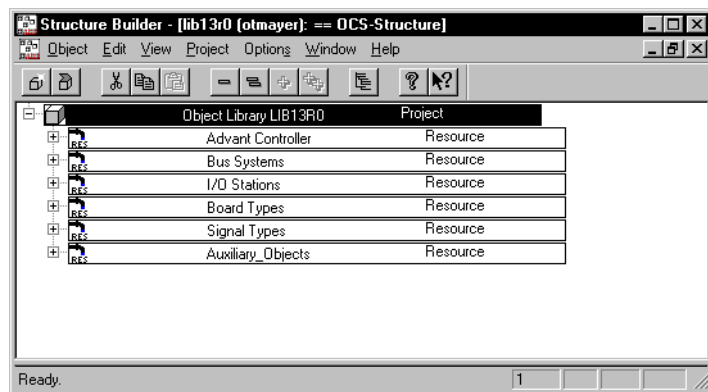


Figure 3-24. Basic Control System Oriented Structure of the Advant Object Library

3.3.1.1 Advant Objects Assigned to Controllers

Figure 3-25 shows Advant object examples that represent Controllers, Bus Systems and Stations assigned to corresponding structure objects:

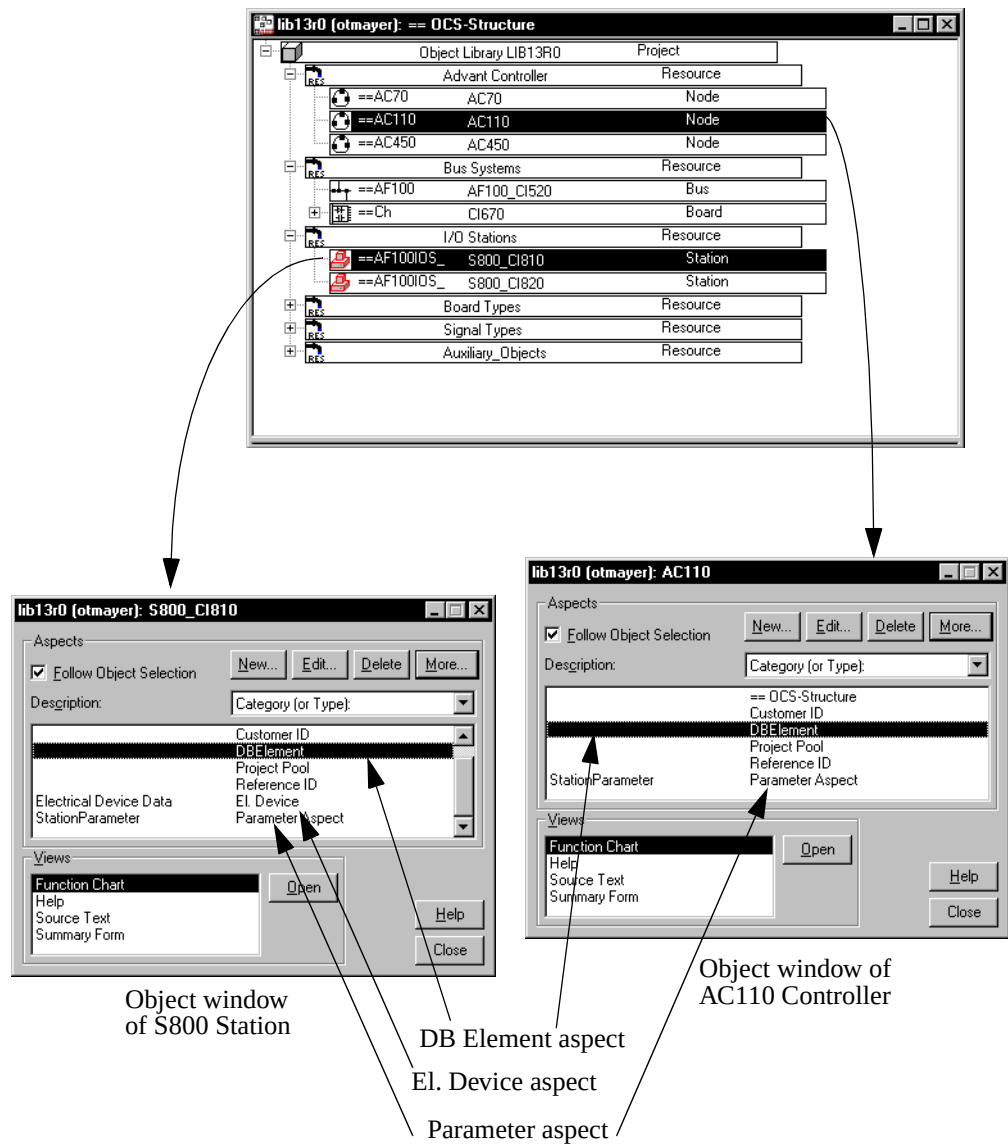


Figure 3-25. Advant Object Types Representing Controllers, Bus Systems and Stations

3.3.1.2 Structure of Boards and Signals

Boards and Signals in the Advant Object Library are classified according to their belonging to the I/O modules S100, S400, S600 and S800.

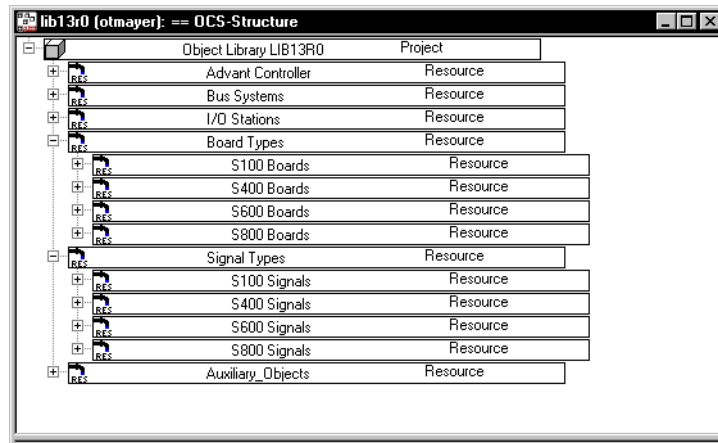


Figure 3-26. Structure of Boards and Signals

3.3.1.3 Board Types in the Advant Object Library

DB Element aspects and Board Parameter aspects are assigned to all board objects in the Advant Object Library.

Board objects are specified with the aspects needed for their use as examples in projects as for example by the import function.

Board channels are assigned to all board objects as children objects.

Examples of S100 I/O Boards

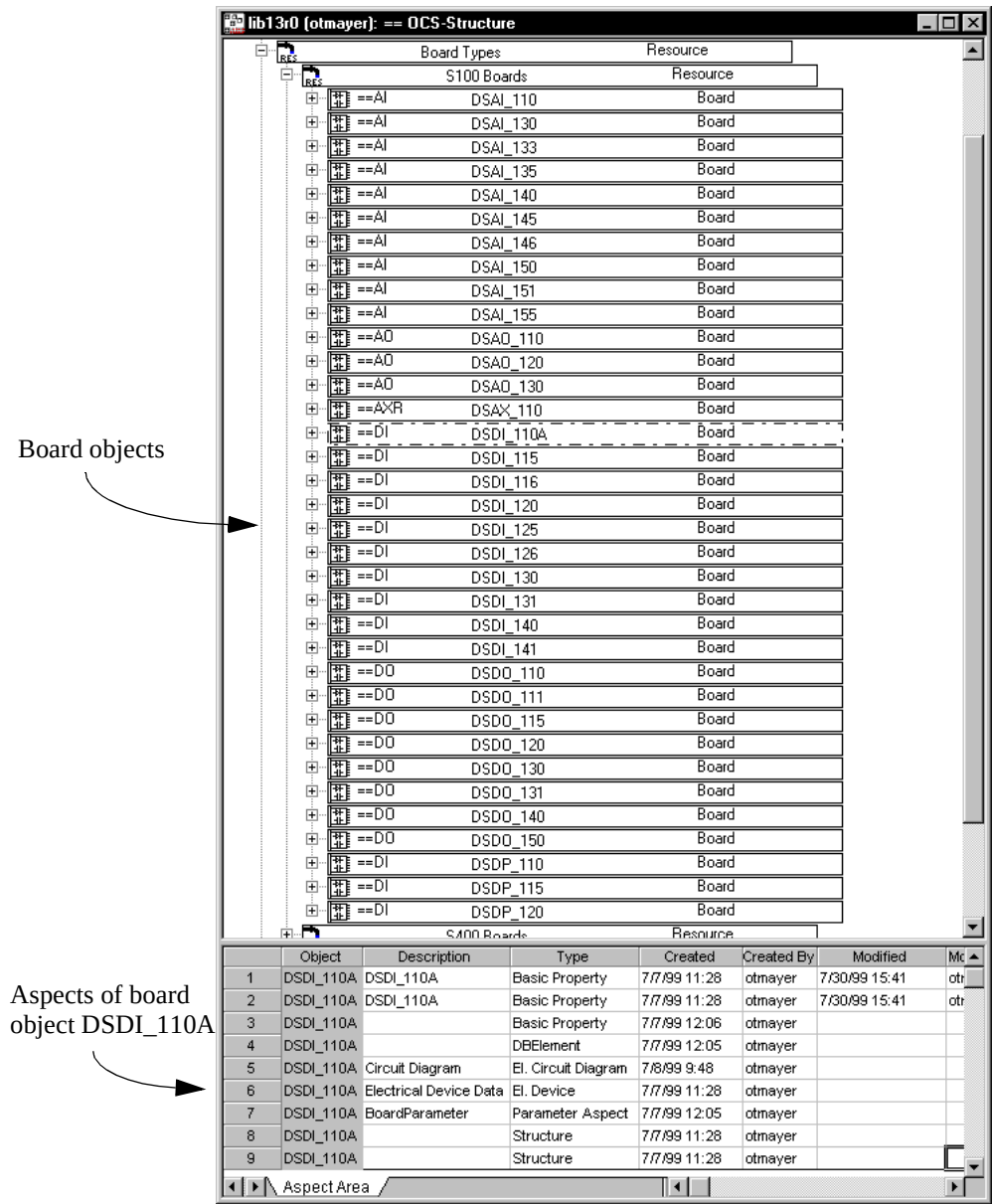


Figure 3-27. S100 I/O Boards

Figure 3-28 shows the Function Chart View on the DB aspect of an Advant object representing

a board. The figure shows the DB section and the Symbol section window.

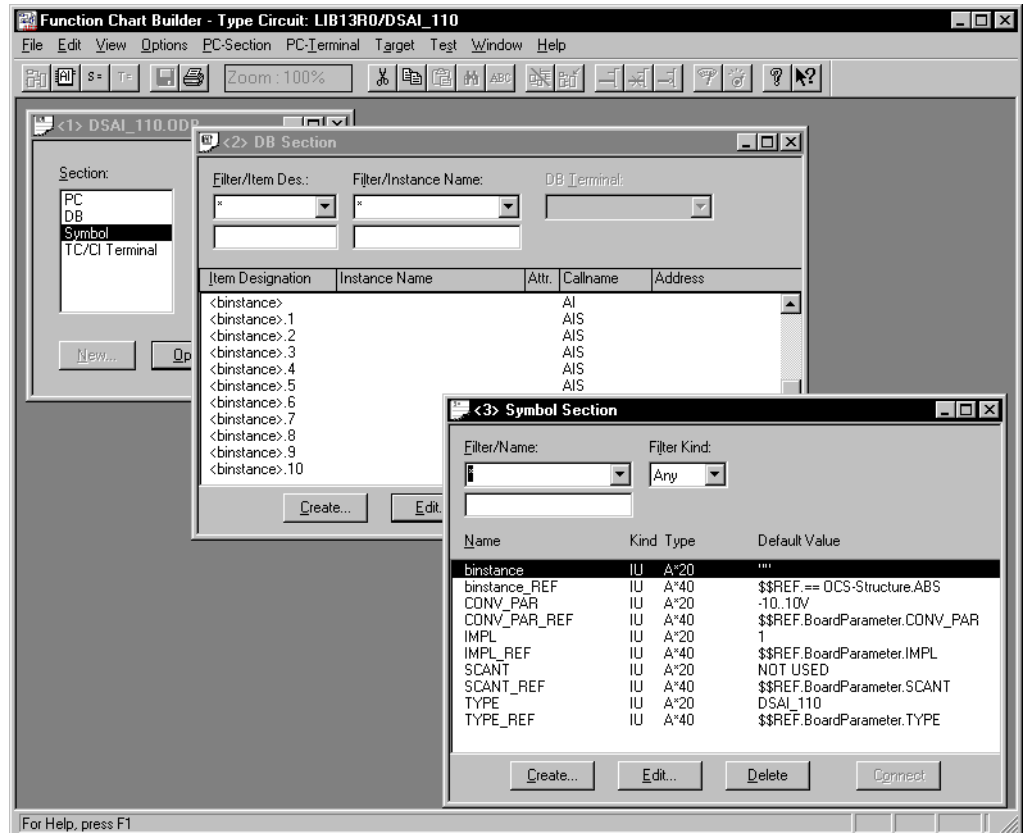


Figure 3-28. Function Chart View on a Board DB Element

Figure 3-29 shows the Attribute view on the parameters of the Board Parameter aspect.

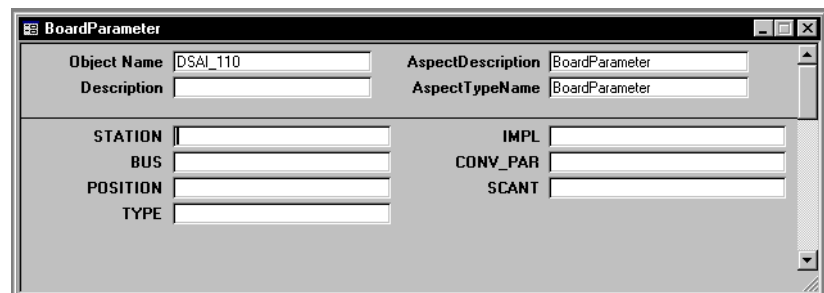
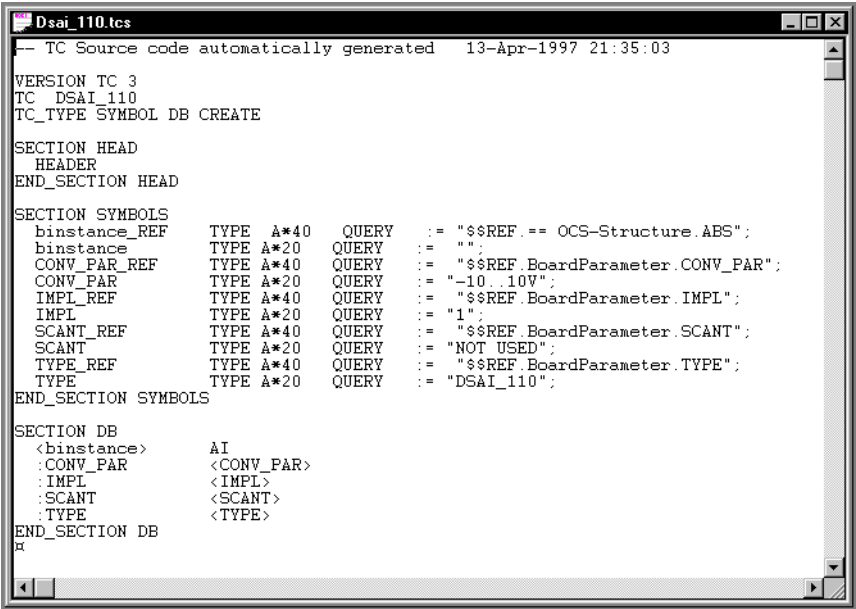


Figure 3-29. Board Parameters

Figure 3-30 shows the Source code of a DB Element aspect of a board object:



```
TC Source code automatically generated 13-Apr-1997 21:35:03

VERSION TC 3
TC DSAI_110
TC_TYPE SYMBOL DB CREATE

SECTION HEAD
HEADER
END_SECTION HEAD

SECTION SYMBOLS
binstance_REF TYPE A*40 QUERY := "$$REF.== OCS-Structure.ABS";
binstance TYPE A*20 QUERY := "";
CONV_PAR_REF TYPE A*40 QUERY := "$$REF.BoardParameter.CONV_PAR";
CONV_PAR TYPE A*20 QUERY := "-10..10V";
IMPL_REF TYPE A*40 QUERY := "$$REF.BoardParameter.IMPL";
IMPL TYPE A*20 QUERY := "1";
SCANT_REF TYPE A*40 QUERY := "$$REF.BoardParameter.SCANT";
SCANT TYPE A*20 QUERY := "NOT USED";
TYPE_REF TYPE A*40 QUERY := "$$REF.BoardParameter.TYPE";
TYPE TYPE A*20 QUERY := "DSAI_110";
END_SECTION SYMBOLS

SECTION DB
<binstance> AI
:CONV_PAR <CONV_PAR>
:IMPL <IMPL>
:SCANT <SCANT>
:TYPE <TYPE>
END_SECTION DB
```

Figure 3-30. Source Code of a Board DB Element

Figure 3-31 shows the Attribute view of the El. Device aspect of a board object. This aspect type is assigned to all board objects in the Advant Object Library and it is set to category IO_CARD.

The category of the El. Device aspect of Board objects is set to IO_CARD

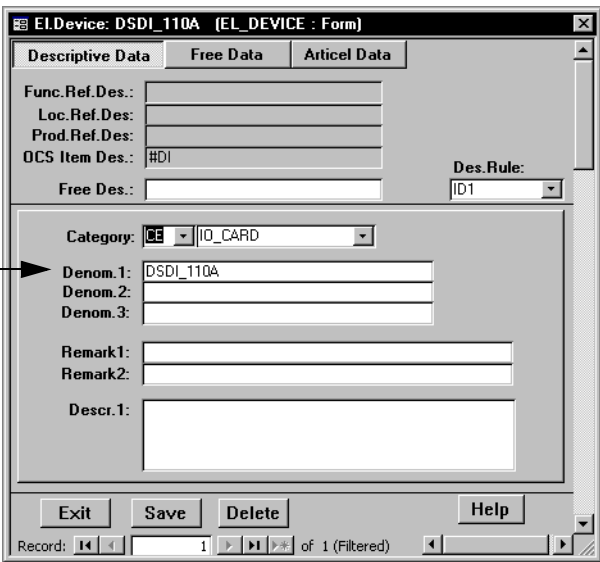


Figure 3-31. Attribute View on an El. Device Aspect of a Board

Figure 3-32 shows the Diagram view of the El. Circuit Diagram aspect of a board object. This aspect defines the relations between board channels and terminals on connection units. This aspects are not yet assigned to all boards (there could be many variants per board) but only to the boards DSDI110A, DSDO115, DI810, DO810. These boards were also provided with connection units as children objects and carry all the necessary information needed to demonstrate a complete engineering workflow.

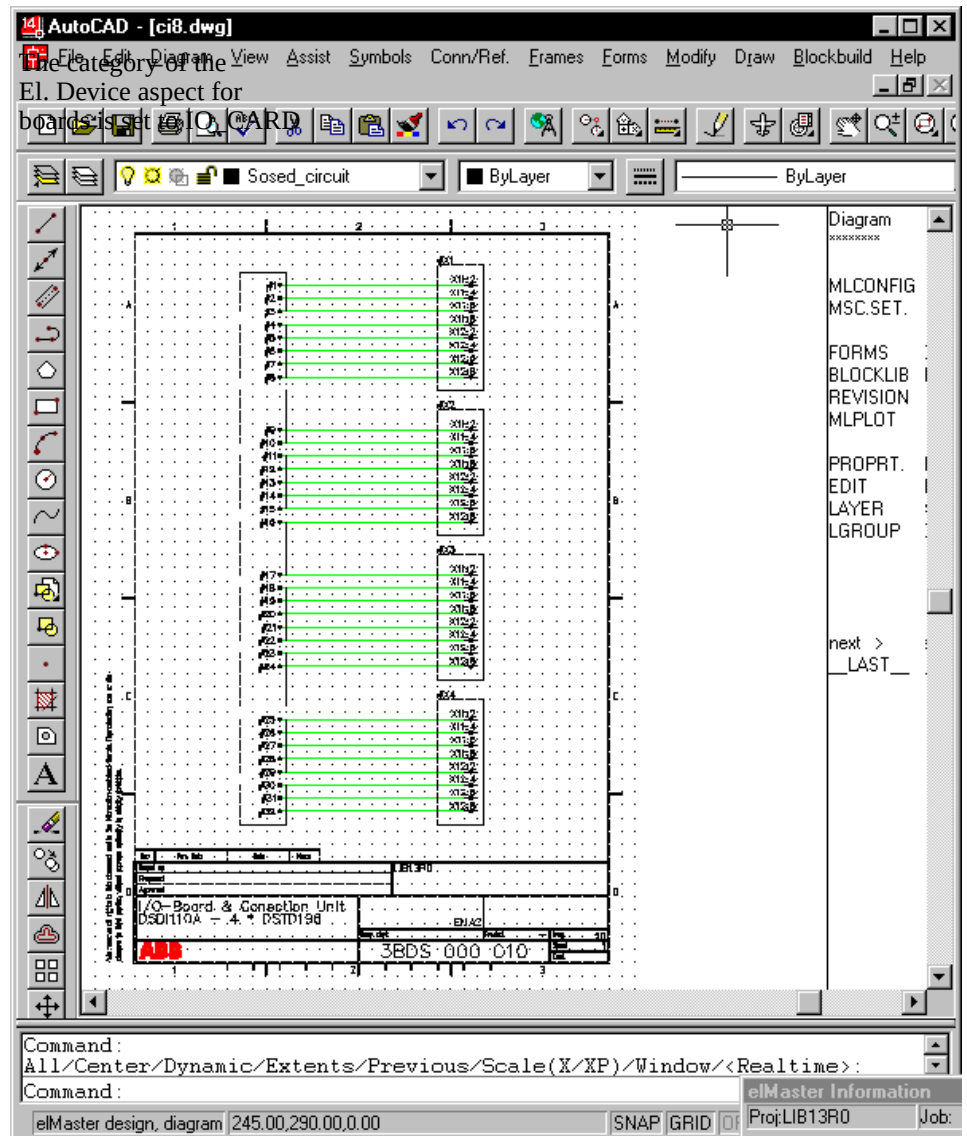


Figure 3-32. Diagram View on the El. Circuit Diagram Aspect of Board DSDI110A

Presentation of Board-Channels in the Advant Object Library

Each channel of a board is represented by a channel object. A reusable relative reference designation is assigned to channel objects.

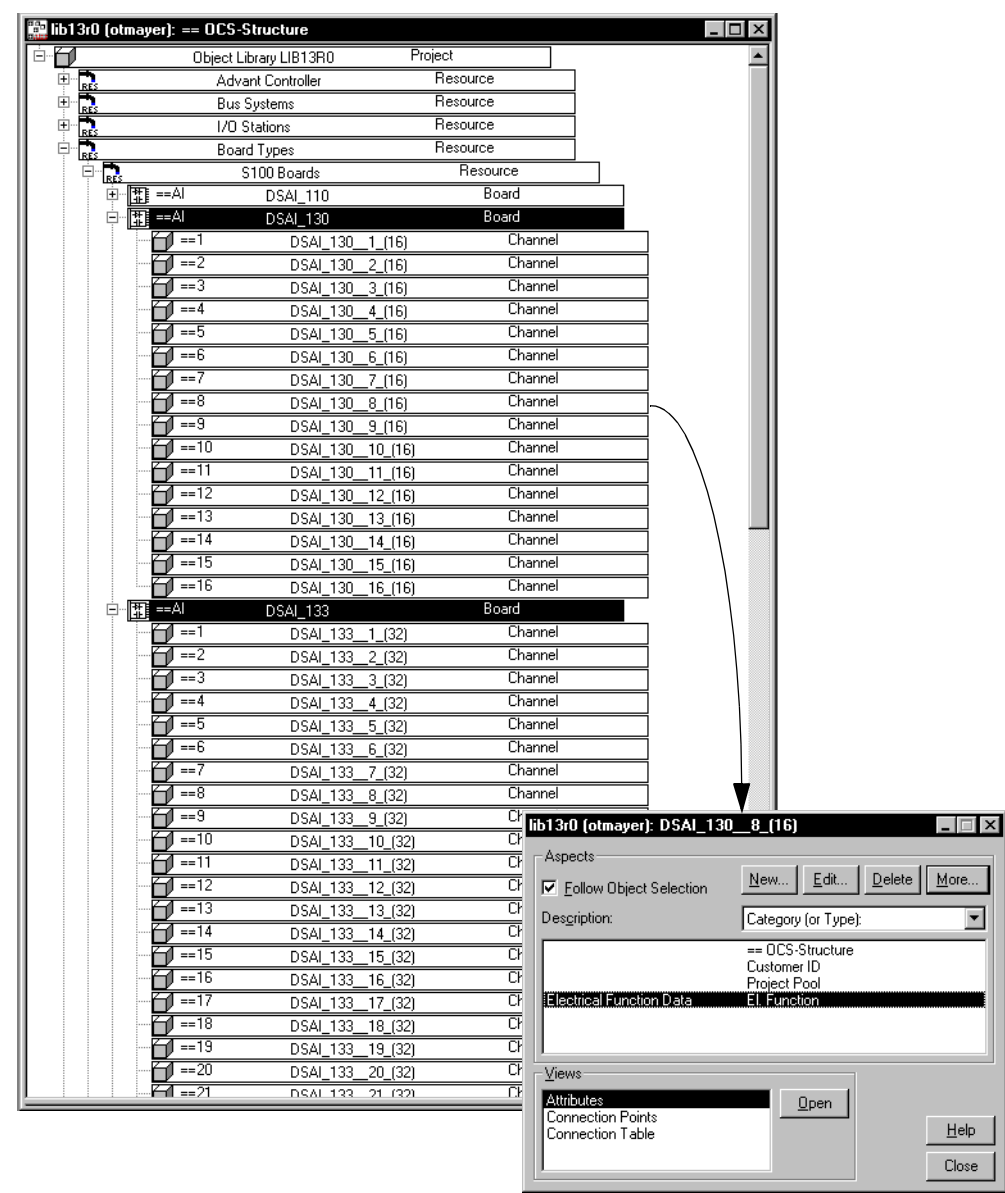


Figure 3-33. Presentation of Board Channels

Figure 3-34 shows the Attribute view of the El. Function aspect of a channel object. This aspect type is assigned to all channel objects in the Advant Object Library and it is set to category IO_CHANNEL.

The category of the El. Function aspect of Channel objects must be set to IO_CHANNEL

Figure 3-34. Attribute View on an El. Function Aspect of a Board

Examples of S400 Boards

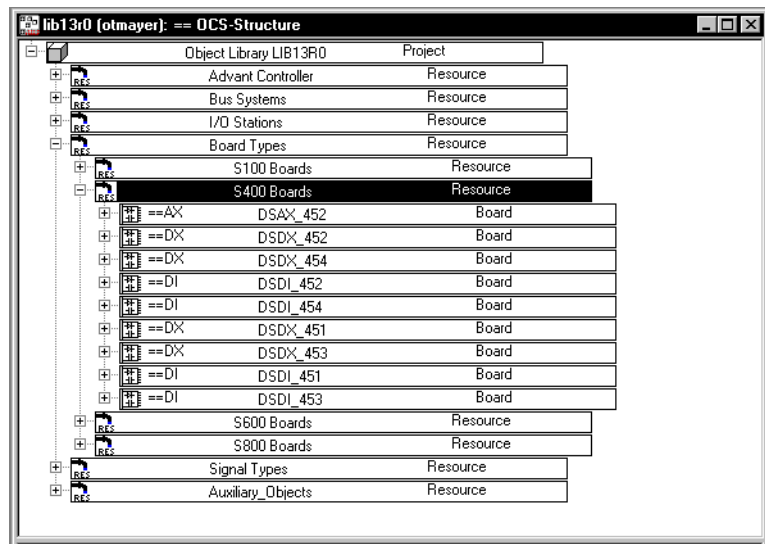


Figure 3-35. S400 I/O Boards

Examples of S600 Boards

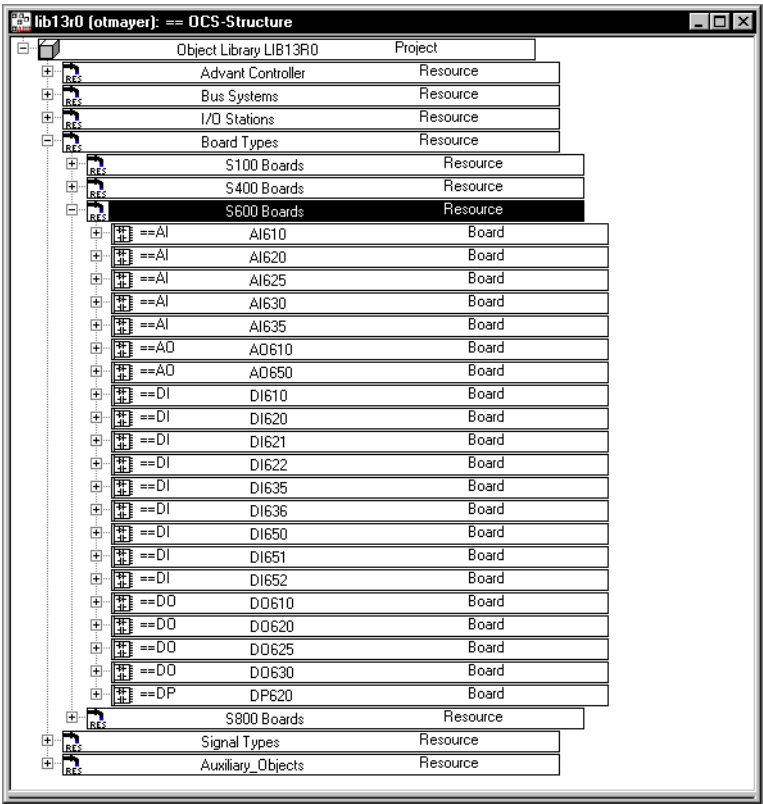


Figure 3-36. S600 I/O Boards

Examples of S800 Boards

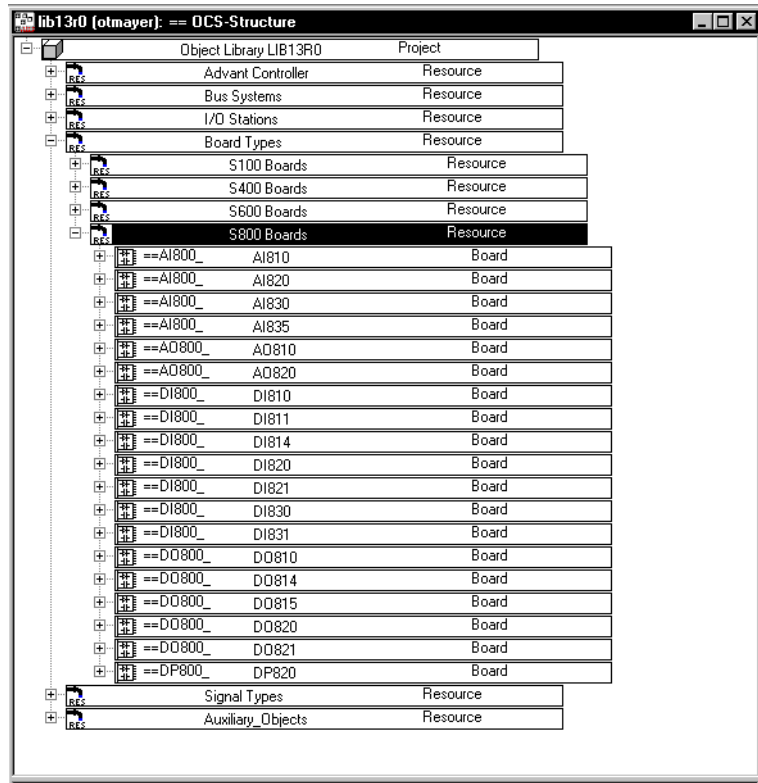


Figure 3-37. S800 I/O Boards

3.3.1.4 Advant Objects Representing Signals

In the OCS-Structure are the same signals placed as in the Function Structure. See the corresponding descriptions in Section 3.2.1.5, Advant Objects Representing Signals.

3.4 Advant Objects Classified as Mounting Locations

Advant objects representing mounting locations are presented in a mounting location oriented structure built up by the Structure Builder. The Advant objects are assigned to a basic structure that groups the objects used as mounting locations into several object classes. The presented basic structure has to be regarded as an example that can be modified and expanded by the user.

3.4.1 Content of the Mounting Location Structure

Advant Object in the Basic Structure

Figure 3-38 shows the basic structure to which objects that represent mounting locations will be assigned:

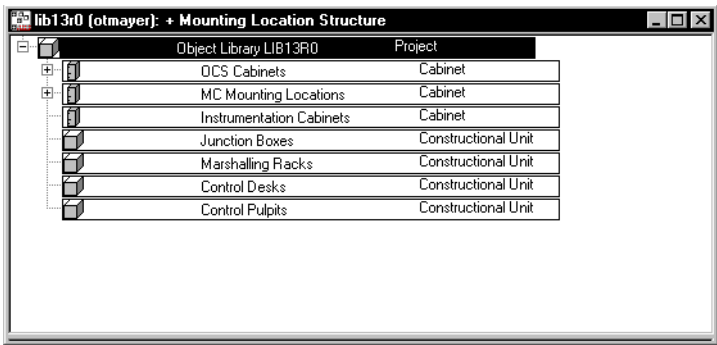


Figure 3-38. Basic Mounting Location Oriented Structure of the Object Library

Types of Cabinet Assemblies

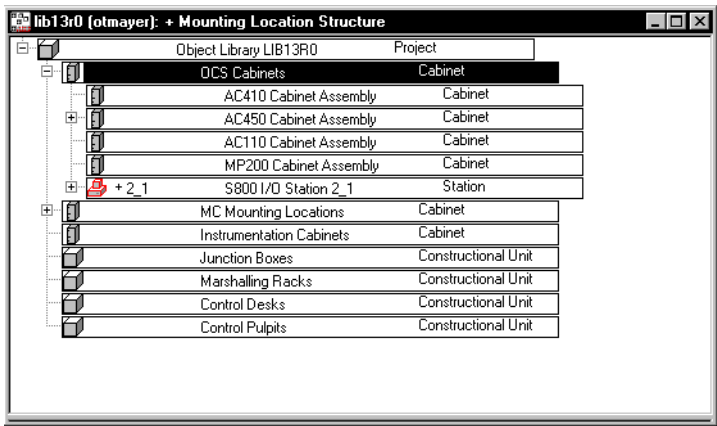


Figure 3-39. OCS-Cabinet Assemblies

Example of Objects representing Racks and Board Locations

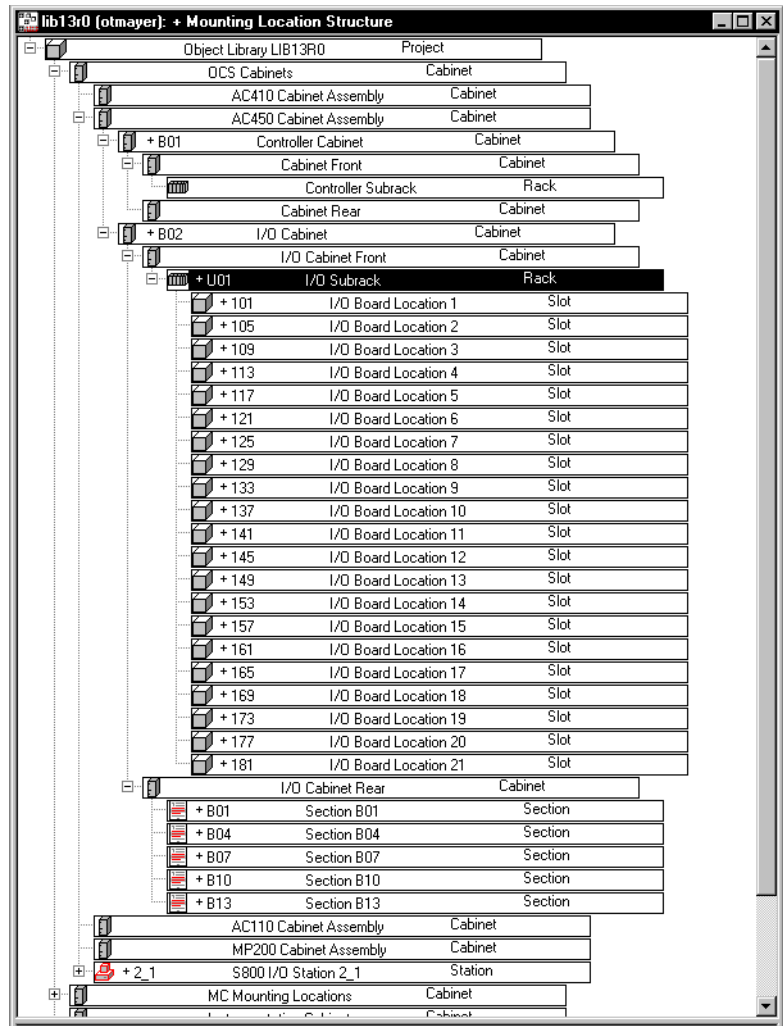


Figure 3-40. Racks and Board Locations in an OCS-Cabinet

Example of Advant objects representing a S800 mounting locations.

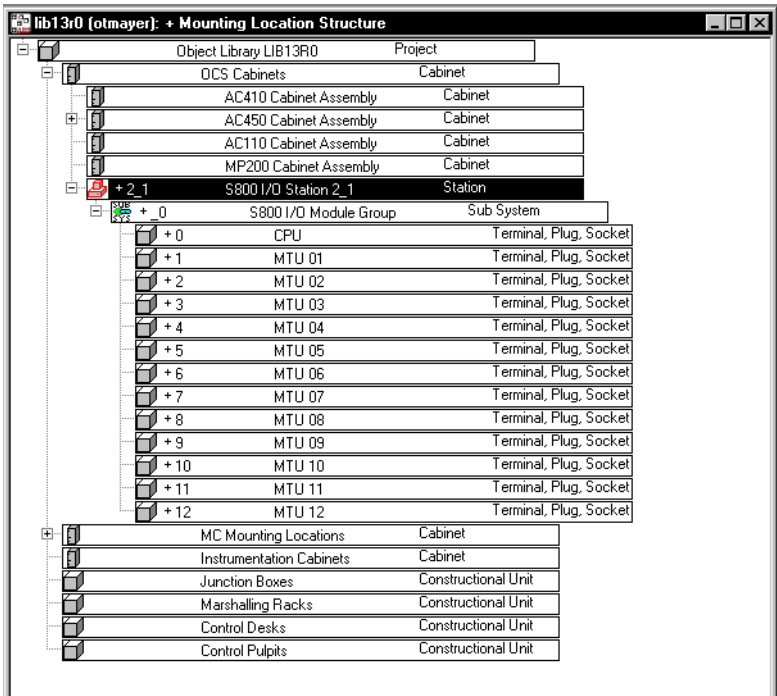


Figure 3-41. Example for a S800 Mounting Unit

Example of Advant objects representing mounting locations for motorstarters

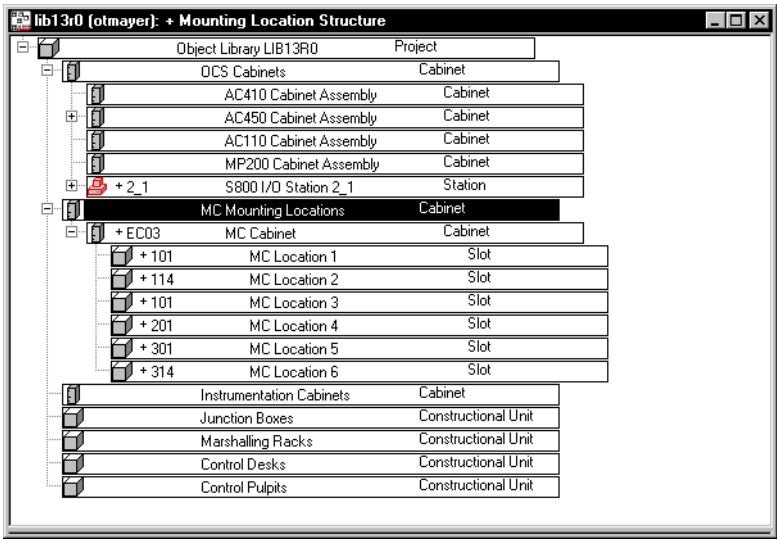


Figure 3-42. Example for Motor Controller Cabinet

Chapter 4 Hints and Recommendations

4.1 Version Settings for Controller and Tools in the Object Library

The Object library contains DB Element aspects, Type Circuit aspects and Circuit aspects. The content of separate libraries is used to create these aspects, that are the DB Element and PC Element libraries. These libraries are accessed by the Function Chart builder for the creation and application of the Object library. The tool versions by which the aspects were created are assigned to the relevant aspects in the Object library.

Only source code files are assigned to OCS relevant aspects in the Object library. Therefore a backtranslate of the Source code has to be done after opening the Function Chart view of an aspect for the first time.

4.1.1 Presettings for DB Elements for Boards and Signals

The *.ini files belonging to the DB element aspects for boards and signals contain the following Builder Version and Node Type settings:

Builder settings in all *.ini files:

- Application Builder Version 2.6/0
- Function Chart Builder Version 4.6/1.

Controller Type settings:

- Node Type AC110 Version 2.3 (for S600 and S800 boards and signals)
- Node Type AC450 with SW QC07-BAS41*3.0 (for S100 and S400 boards and signals).

4.1.2 Changing the Version Settings

4.1.2.1 Changing the Controller Version

You can change the controller version used for a DB Element by the following procedure:

1. Open the Object window of the object with the relevant DB Element aspect.
2. Open the **Summary** view of the **DB Element** aspect.
3. Select the new node type and /or version in the **Node Type** combo box of the **Modify Type Circuit** window.
4. Click on **OK**.

Figure 4-1 shows the Modify Type Circuit window that represents the Summary view of a DB Element aspect:

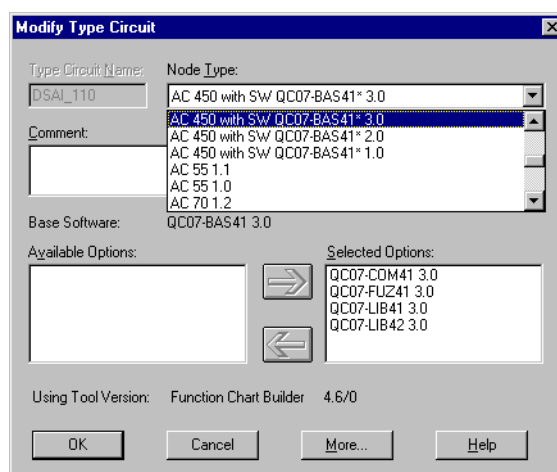


Figure 4-1. Modify Type Circuit Window

4.1.2.2 Changing the FCB Version

You can change the FCB version assigned to a DB Element by the following procedure:

1. Open the Object window of the object with the relevant DB Element aspect.
2. Open the **Summary** view of the **DB Element** aspect.
3. Click on **More** in the **Modify Type Circuit** window.
4. Select **Tool Versions** in the **More Settings** box.
5. Select Function Chart Builder version in the **Select Tool Versions** box.
6. Click on **OK**.

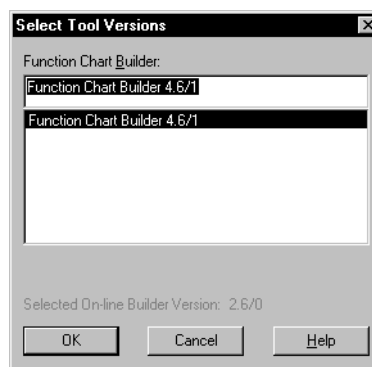


Figure 4-2. Select Tool Versions Box

4.2 Recommendations

4.2.1 Configuring an Object with Circuit Aspect Containing Type Circuit

7. Configure the type circuit. Use allowed parameter references.

Make sure that the actual values of symbols are supplied either by default values or by parameter references. Actual values supplied during instantiation into the circuit or by editing symbol values in the TC instances will get lost in Drop and UpdateParameter.

For symbol controlled default connections of output terminals to DB element terminals make sure that the = sign is outside of any symbol used in the default connection specification.

8. Configure the Circuit aspect.
 - a. Introduce the circuit instance symbol finstance.
 - b. Begin the circuit of this aspect with a FUNCN (or another structure element) and give this element an instance name defined by the circuit's instance symbol finstance (INAME <finstance>).
 - c. Into the circuit below the structure element instantiate the type circuit(s). Make sure before you instantiate that the symbol substitution in FCB circuit mode is switched off.

Note that the TC instances do not need to get an instance name, instance names are given to the TC instances below the structure element automatically.

Note that only one Circuit aspect per object is supported. Additional Circuit aspects are ignored during drag and drop without giving notice.

Note that the declaration length of the finstance symbol must fit to the length of its contents generated from basic property name of category Customer ID concatenated with_(underscore) and the aspect description of the Circuit aspect.

Note that you can also configure a circuit without using a structure element. Then only one TC instance is allowed and this TC instance has to get the instance name <finstance>.

If the process object shall be used with the Locate Circuit command, then it must be placed in the OCS Structure below the node object it will be inserted into, and the relative reference designation entry in the OCS Structure aspect of the process object must contain the PC item designation of the insertion point PC element in the node.

4.2.2 Configuring an Object to Create DB Elements by Drag and Drop

1. Configure the DB type circuit. Rules except the connection-related ones (DB and PC connections), see Section 4.2.1 and in *Application Builder Release Notes*.
2. Configure the Circuit aspect
 - a. Introduce the circuit instance symbol finstance.
 - b. Instantiate the DB TC into the circuit using an instance name defined by the circuit's instance symbol finstance (INAME <finstance>).

Note that only one DB type circuit per object is supported regarding parameter update.

Note that only one Circuit aspect per object is supported. Additional Circuit aspects are ignored during drag and drop without notification..

4.2.3 Configuring an Object to Create DB Elements by Generate DB

1. Configure the DB type circuit. Rules except the connection-related ones (DB and PC connections), see Section 4.2.1 and in *Application Builder Release Notes*.
2. Configure the DBElement aspect by attaching the type circuit or/and editing it.

Note that instance naming for the DB TC instance is done automatically when using the DBElement aspect with a DBSection aspect.

4.2.4 Setting Options for Bulk Copy From Library

Before doing a bulk copy of objects from library project to another target project (e.g. before import or before performing manually a copy with multiple selection) **set the following options for the target project** by using the SetOptions command on the target project's Project OCS Defaults aspect:

- Bulk Data
- Generate TCName
- Generate CName
- Generate NodeName.

This switches off the Copy dialog for the aspects owned by the Control Configuration Aspect System during the bulk copy process and generates file-object names automatically.

Switching on the **Generate Name** options is only needed if the destination project already contains instances with the same names as in the source project.

If you copied between projects without having switched on the corresponding **Generate Name** options, you should not copy the same objects again without either switching off the **Bulk Data** option or switching on additionally the **Generate Name** option. Otherwise an error message is displayed, caused by the fact that the objects already exist in your project. The same applies to trying to copy within the project with only the **Bulk Data** option switched on.

4.2.5 DBElement Aspect Copied from Library

In the target project this aspect cannot be opened as it contains a reference to a library TC on the directory specified by the TC\$LEV1 or TC\$LEV2 environment variable. A corresponding message is displayed if you try to do so.

Note that such aspects can be copied nevertheless.

If you need the TC project specific in your target project you have to copy all files for this TC from the project specific TC directory of the library project to the project specific TC directory of your target project manually after having copied the object containing the DBElement aspect.

4.2.6 Prepare Structure Aspects

Make sure before using objects from the OCS- structure of the Object Library that

- the == prefix is set in the OCS Structure aspect of all objects
- the < control for the relative reference designation is set in the OCS Structure aspect of all channels.

4.2.7 Environment Variables TC\$LEV1 and TC\$LEV2

For using type circuits of a library project the environment variable TC\$LEV1 has to be set to the project specific TC directory of the Object Library.

Example: TC\$LEV1=L:\DATA\LIB13R0\NODES\PR_TC\TCDATA

The setup procedure for the Advant Object library performs a default setting to this directory of the library project.

If you want to reference type circuits in a second library project set TC\$LEV2 accordingly.



3BSE 013 934R301

September 1999